A Bayesian framework for word segmentation: Exploring the effects of context

Goldwater, S., Griffiths, T. L., & Johnson, M. (2009). A Bayesian framework for word segmentation: Exploring the effects of context. Cognition, 112(1), 21-54doi: 10.1016/j.cognition.2009.03.008

A Bayesian Framework for Word Segmentation: Exploring the Effects of Context

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson

Preprint for publication in *Cognition*

Abstract

Since the experiments of Saffran et al. (1996a), there has been a great deal of interest in the question of how statistical regularities in the speech stream might be used by infants to begin to identify individual words. In this work, we use computational modeling to explore the effects of different assumptions the learner might make regarding the nature of words – in particular, how these assumptions affect the kinds of words that are segmented from a corpus of transcribed child-directed speech. We develop several models within a Bayesian ideal observer framework, and use them to examine the consequences of assuming either that words are independent units, or units that help to predict other units. We show through empirical and theoretical results that the assumption of independence causes the learner to undersegment the corpus, with many two- and three-word sequences (e.g. *what's that*, *do you*, *in the house*) misidentified as individual words. In contrast, when the learner assumes that words are predictive, the resulting segmentation is far more accurate. These results indicate that taking context into account is important for a statistical word segmentation strategy to be successful, and raise the possibility that even young infants may be able to exploit more subtle statistical patterns than have usually been considered.

1 Introduction

One of the first problems infants must solve as they are acquiring language is word segmentation: identifying word boundaries in continuous speech. About 9% of utterances directed at English-learning infants consist of isolated words (Brent and Siskind, 2001), but there is no obvious way for children to know from the outset which utterances these are. Since multi-word utterances generally have no apparent pauses between words, children must be using other cues to identify word boundaries. In fact, there is evidence that infants use a wide range of weak cues for word segmentation. These cues include phonotactics (Mattys et al., 1999), allophonic variation (Jusczyk et al., 1999a), metrical (stress) patterns (Morgan et al., 1995; Jusczyk et al., 1999b), effects of coarticulation (Johnson and Jusczyk, 2001), and statistical regularities in the sequences of syllables found in speech (Saffran et al., 1996a). This last source of information can be used in a language-independent way, and seems to be used by infants earlier than most other cues, by the age of 7 months (Thiessen and Saffran, 2003). These facts have caused some researchers to propose that strategies based on statistical sequencing information are a crucial first step in bootstrapping word segmentation (Thiessen and Saffran, 2003), and have provoked a great deal of interest in these strategies (Saffran et al., 1996b; Saffran et al., 1996a; Aslin et al., 1998; Toro et al., 2005). In this paper, we use computational modeling techniques to examine some of the assumptions underlying much of the research on statistical word segmentation.

Most previous work on statistical word segmentation is based on the observation that transitions from one syllable or phoneme to the next tend to be less predictable at word boundaries

than within words (Harris, 1955; Saffran et al., 1996a). Behavioral research has shown that infants are indeed sensitive to this kind of predictability, as measured by statistics such as transitional probabilities (Saffran et al., 1996a; Aslin et al., 1998). This research, however, is agnostic as to the mechanisms by which infants use statistical patterns to perform word segmentation. A number of researchers in both cognitive science and computer science have developed algorithms based on transitional probabilities, mutual information, and similar statistics of predictability in order to clarify how these statistics can be used procedurally to identify words or word boundaries (Swingley, 2005; Ando and Lee, 2000; Feng et al., 2004; Cohen and Adams, 2001). Here, we take a different approach: we seek to identify the *assumptions* the learner must make about the nature of language in order to correctly segment natural language input.

Observations about predictability at word boundaries are consistent with two different kinds of assumptions about what constitutes a *word*: either a word is a unit that is statistically independent of other units, or it is a unit that helps to predict other units (but to a lesser degree than the beginning of a word predicts its end). In most artificial language experiments on word segmentation, the first assumption is adopted implicitly by creating stimuli through random (or near-random) concatenation of nonce words. This kind of random concatenation is often necessary for controlled experiments with human subjects, and has been useful in demonstrating that humans are sensitive to the statistical regularities in such randomly generated sequences. However, it obviously abstracts away from many of the complexities of natural language, where regularities exist not only in the relationships between sub-word units, but also in the relationships between words themselves. We know that humans are able to use sub-word regularities to begin to extract words; it is natural to ask whether attending to these kinds of regularities is sufficient for a statistical learner to succeed with word segmentation in a more naturalistic setting. In this paper, we use computer simulations to examine learning from natural, rather than artificial, language input. We ask what kinds of words are identified by a learner who assumes that words are statistically independent, or (alternatively) by a learner who assumes as well that words are predictive of later words. We investigate this question by developing two different Bayesian models of word segmentation incorporating each of these two different assumptions. These models can be seen as *ideal learners*: they are designed to behave optimally given the available input data, in this case a corpus of phonemically transcribed child-directed speech.

Using our ideal learning approach, we find in our first set of simulations that the learner who assumes that words are statistically independent units tends to undersegment the corpus, identifying commonly co-occurring sequences of words as single words. These results seem to conflict with those of several earlier models (Brent, 1999; Venkataraman, 2001; Batchelder, 2002), where systematic undersegmentation was not found even when words were assumed to be independent. However, we argue here that these previous results are misleading. Although each of these learners is based on a probabilistic model that defines an optimal solution to the segmentation problem, we provide both empirical and analytical evidence that the segmentations found by these learners are not the optimal ones. Rather, they are the result of limitations imposed by the particular learning algorithms employed. Further mathematical analysis shows that undersegmentation is the optimal solution to the learning problem for *any* reasonably defined model that assumes statistical independence between words.

Moving on to our second set of simulations, we find that permitting the learner to gather information about word-to-word dependencies greatly reduces the problem of undersegmentation. The corpus is segmented in a much more accurate, adult-like way. These results indicate that, for an ideal learner to identify words based on statistical patterns of phonemes or syllables, it is important to take into account that frequent predictable patterns may occur *either* within words *or* across words. This kind of dual patterning is a result of the hierarchical structure of language, where predictable patterns occur at many different levels. A learner who considers predictability at only one level (sub-word units within words) will be less successful than

a learner who considers also the predictability of larger units (words) within their sentential context. The second, more nuanced interpretation of the statistical patterns in the input leads to better learning.

Our work has important implications for the understanding of human word segmentation. We show that successful segmentation depends crucially on the assumptions that the learner makes about the nature of words. These assumptions constrain the kinds of inferences that are made when the learner is presented with naturalistic input. Our ideal learning analysis allows us to examine the kinds of constraints that are needed to successfully identify words, and suggests that infants or young children may need to account for more subtle statistical effects than have typically been discussed in the literature. To date, there is little direct evidence that very young language learners approximate ideal learners. Nevertheless, this suggestion is not completely unfounded, given the accumulating evidence in favor of humans as ideal learners in other domains or at other ages (Xu and Tenenbaum, 2007; Frank et al., 2007; Schulz et al., 2007). In order to further examine whether infants behave as ideal learners, or the ways in which they depart from the ideal, it is important to first understand what behavior to expect from an ideal learner. The theoretical results presented here provide a characterization of this behavior, and we hope that they will provide inspiration for future experimental work investigating the relationship between human learners and ideal learners.

The remainder of this paper is organized as follows. First, we briefly review the idea of transitional probabilities and how they relate to the notion of words, and provide some background on the probabilistic modeling approach taken here. We draw a distinction between two kinds of probabilistic model-based systems – those based on *maximum-likelihood* and *Bayesian* estimation – and argue in favor of the Bayesian approach. We discuss in some detail the strengths and weaknesses of the Model-Based Dynamic Programming (MBDP-1) system, a Bayesian learner described by Brent (1999). Next, we introduce our own Bayesian unigram model and learning algorithm, which address some of the weaknesses of MBDP-1. We provide the results of simulations using this model and compare them to the results of previously proposed models. We then generalize our unigram modeling results using additional empirical and theoretical arguments, revealing some deep mathematical similarities between our unigram model and MBDP-1. Finally, we extend our model to incorporate bigram dependencies, present the results of this bigram model, and conclude by discussing the implications of our work.

2 Words and transitional probabilities

The question of how infants begin to segment words from continuous speech has inspired a great deal of research over the years (Jusczyk, 1999). While many different cues have been shown to be important, here we focus on one particular cue: statistical regularities in the sequences of sounds that occur in natural language. The idea that word and morpheme boundaries may be discovered through the use of statistical information is not new, but originally these methods were seen primarily as analytic tools for linguists (Harris, 1954; Harris, 1955). More recently, evidence that infants are sensitive to statistical dependencies between syllables has lent weight to the idea that this kind of information may actually be used by human learners for early word segmentation (Saffran et al., 1996a; Thiessen and Saffran, 2003). In particular, research on statistical word segmentation has focused on the notion of *transitional probabilities* between sub-word units (e.g., segments or syllables). The transitional probability from (say) syllable x to syllable y is simply the conditional probability of y given x. In natural language, there is a general tendency towards lower transitional probabilities at word boundaries than within words (Harris, 1954; Saffran et al., 1996b), a tendency which infants seem able to exploit in order to segment word-like units from continuous speech (Saffran et al., 1996a; Aslin et al., 1998). While other cues are also important for word segmentation, and may in fact take precedence over transitional probabilities in older infants, transitional probabilities seem to be one of the

earliest cues that infants are able to use for this task (Johnson and Jusczyk, 2001; Thiessen and Saffran, 2003).

Much of the experimental work devoted to studying word segmentation and related linguistic tasks has focused on exploring the kinds of statistical information that human learners are or are not sensitive to (e.g., transitional probabilities vs. frequencies (Aslin et al., 1998), syllables vs. phonemes (Newport et al., in preparation), adjacent vs. non-adjacent dependencies (Newport and Aslin, 2004), and the ways in which transitional probabilities interact with other kinds of cues (Johnson and Jusczyk, 2001; Thiessen and Saffran, 2003; Thiessen and Saffran, 2004). In addition, many researchers have explored the extent to which word segmentation based on transitional probabilities can be viewed as a special case of more general pattern- or sequence-learning mechanisms that operate over a range of cognitive domains (Fiser and Aslin, 2002; Creel et al., 2004). A question that has received less explicit attention is how the notion of transitional probabilities relates to the notion of words. Transitional probabilities are a property of the boundaries between words (or units within words), but ultimately it is the words themselves, rather than the boundaries, that are of interest to the language learner/user. It behooves us, then, to consider what possible properties of words (or, more accurately, word sequences) could give rise to the patterns of transitional probabilities that are typically discussed in the literature, i.e. lower probabilities at word boundaries and higher probabilities within words.

Given a lexicon of words, one way that the standard patterns of transitional probabilities can arise is by choosing words independently at random from the lexicon and stringing them together to form a sequence. To a first approximation, this is the procedure that is typically used to generate stimuli for most of the experiments mentioned above.[1] There are many good reasons to generate experimental stimuli in this way, especially when the focus of research is on transitional probabilities: choosing words at random controls for possible ordering effects and other confounds, and leads to simple and systematic patterns of transitional probabilities. However, there is clearly another way we could generate a sequence of words, by choosing each word conditioned on the previous word or words. Depending on the strength of the word-to-word dependencies, transitional probabilities between words may be low or high. In general, if the strength of the dependencies between words is variable, then in a non-independent sequence of words, word boundaries will still tend to be associated with lower transitional probabilities (since many pairs of words will not be highly dependent). However, there will also be word boundaries with relatively high transitional probabilities (where two words are highly associated, as in *rubber ducky* or *that's a*).

The models that we develop in this paper are designed to examine the results of making these two different assumptions about the nature of language: that words are statistically independent units, or that they are predictive units. In thinking about the differences between learners making each of these two kinds of assumptions, we frame the issue in terms of the space of linguistic hypotheses (loosely, grammars) that each learner considers. Notice that a learner who assumes that utterances are formed from sequences of independently chosen words is more restricted than a learner who assumes that words may predict other words. The second learner is able to learn grammars that describe either predictive or non-predictive sequences of words, while the first learner can only learn grammars for non-predictive sequences of words. If words are truly independent, then the first learner may have an advantage due to the presence of the stronger constraint, because this learner has a much smaller space of hypotheses to consider. On the other hand, the second learner will have an advantage in the case where words are not independent, because the learner who assumes independence will never be able to converge on the correct hypothesis. Before describing our implementation of these two kinds of learners, we

[1]The words in experimental stimuli are never chosen completely independently, due to restrictions against immediate repetition of words. When the lexicon is small, this leads to significant deviations from independence. However, as the lexicon size grows, sequences without repetition will become more and more similar to truly independent sequences.

first outline our general approach and provide a summary of related work.

2.1 Probabilistic models for word segmentation

Behavioral work in the vein of Saffran et al. (1996a) has provided a wealth of information regarding the kinds of statistics human learners are sensitive to, at what ages, and to what degree relative to other kinds of segmentation cues. Computational modeling provides a complementary method of investigation that can be used to test specific hypotheses about *how* statistical information might be used procedurally to identify word boundaries or *what* underlying computational problem is being solved. Using the terminology of Marr (1982), these two kinds of questions can be investigated by developing models at (respectively) the *algorithmic* level or *computational* level of the acquisition system. Typically, researchers investigate algorithmic-level questions by implementing algorithms that are believed to incorporate cognitively plausible mechanisms of information processing. Algorithmic-level approaches to word segmentation include a variety of neural network models (Elman, 1990; Allen and Christiansen, 1996; Cairns and Shillcock, 1997; Christiansen et al., 1998) as well as several learning algorithms based on transitional probabilities, mutual information, and similar statistics (Swingley, 2005; Ando and Lee, 2000; Feng et al., 2004; Cohen and Adams, 2001) (with most of the latter group coming from the computer science literature).

In contrast to these proposals, our work provides a computational-level analysis of the word segmentation problem. A computational-level approach focuses on identifying the problem facing the learner and determining the logic through which it can be solved. For problems of induction such as those facing the language learner, probability theory provides a natural framework for developing computational-level models. A *probabilistic model* is a set of declarative mathematical statements specifying the goals of the learning process and the kinds of information that will be used to achieve them. Of course, these declarative statements must be paired with some algorithm that can be used to achieve the specific goal, but generally the algorithm is not seen as the focus of research. Rather, computational-level investigations often take the form of *ideal learner* analyses, examining the behavior of a learner who behaves optimally given the assumptions of the model.[2]

Very generally, we can view the goal of a language learner as identifying some abstract representation of the observed data (e.g., a grammar) that will allow novel linguistic input to be correctly interpreted, and novel output to be correctly produced. Many different representations are logically possible, so the learner must have some way to determine which representation is most likely to be correct. Probabilistic models provide a natural way to make this determination, by creating a probability distribution over different hypothesized representations given the observed data. A learner who is able to correctly identify this *posterior* distribution over hypotheses can use this information to process future input and output in an optimal way (i.e., in a way that is as similar as possible to the correct generating process — the adult grammar). Under this view, then, the posterior distribution over grammars is the outcome of the learning process.

How does the learner go about identifying the posterior distribution? Bayes' rule tells us that the probability of a hypothesized grammar h given the observed data d can be computed

[2] A note on terminology: the word *model* unfortunately encompasses two related but (importantly) distinct senses. It can be used to describe either (1) a proposal about the nature of learning or its implementation (as in "connectionist model", "exemplar model"); or (2) a specific mathematical statement regarding the process generating a set of data (as in "probabilistic model", "generative model"). A probabilistic model (second sense) together with its learning algorithm can be viewed as an instance of a learning model (first sense). To avoid confusion, we will generally use the term "model" only for the second sense, and the terms "system" or "learner" to describe the fully implemented combination of a probabilistic model and learning algorithm.

as

$$P(h|d) = \frac{P(d|h)P(h)}{\sum_{h'} P(d|h')P(h')}$$

where the sum in the denominator ranges over all hypotheses h' within the *hypothesis space*. Here, $P(d|h)$ (known as the *likelihood*) is the probability of the observed data given a particular hypothesis, and tells us how well that hypothesis explains the data. $P(h)$ (the *prior* probability of h) tells us how good a linguistic hypothesis h is, regardless of any data. The prior can be viewed as a learning bias or measure of linguistic naturalness: hypotheses with high prior probability may be adopted based on less evidence than hypotheses with low prior probability. Bayes' rule states that the posterior probability $P(h|d)$ is proportional to the product of the likelihood and the prior, with the denominator acting as a normalizing constant to ensure that $P(h|d)$ sums to one over all hypotheses. The learner can compute the posterior probabilities of different hypotheses by evaluating each one according to its explanatory power (likelihood) and the learner's prior expectations.

Bayes' rule answers the question of how to determine the posterior probability of a hypothesis if the prior probability and likelihood are known, but it does not tell us how to compute those terms in the first place. We turn first to the calculation of the likelihood. Typically, the likelihood is computed by defining a *generative model*: a probabilistic process for generating the observed data given the hypothesis under consideration. As a simple non-linguistic example, imagine the data d consists of the results of 10 coin flips and we wish to determine the probability of d given hypotheses h that differ in the probability of flipping heads. We assume a generative model in which each observation is the result of flipping the same coin, and that coin always has probability p of landing on heads independently of any previous outcomes. The set of hypotheses under consideration consists of all possible values for p. We therefore have $P(d \mid h = p) = p^{n_H}(1-p)^{n_T}$, where n_H and n_T are the number of heads and tails observed in a particular sequence of flips.

2.1.1 Maximum-likelihood estimation

A standard method of using a probabilistic generative model for learning is to perform *maximum-likelihood estimation*, i.e., to choose the hypothesis $\hat{h}$ that maximizes the value of the likelihood function. This is equivalent to choosing the hypothesis with maximum posterior probability, assuming a uniform prior distribution over hypotheses with respect to the given parameterization. In the coin flip example, it is easy to show using elementary calculus that the likelihood is maximized when $h = \frac{n_H}{n_H+n_T}$. Thus, if we observe a sequence consisting of four tails and six heads, the maximum-likelihood estimate for h is $\hat{h} = .6$.

For more complex generative models such as those typically used in language modeling, it is usually impossible to identify the maximum-likelihood hypothesis analytically. If the number of possible hypotheses is small, it may be feasible to explicitly compute the likelihood for each possible hypothesis and choose the best one. However, in general, it will be necessary to design some sort of algorithm for searching through the space of hypotheses without evaluating all of them. The ideal algorithm would be one that is guaranteed to find the globally optimal hypothesis. In many cases, however, *approximate* search algorithms are used. These algorithms generally work by seeking the optimal hypothesis within some local region of the search space. Approximate search algorithms may be used for practical reasons (when an exact procedure is not known, as in Anderson (1991)) or for theoretical reasons (if the researcher wishes to incorporate particular assumptions about human learning, as in Sanborn et al. (2006b)). In either case, certain hypotheses are excluded from consideration by the algorithm itself. Consequently, the use of an approximate search procedure can make a purely computational-level analysis difficult. The kinds of generalizations made by the learner are determined both by the explicit constraints

specified by the probabilistic model, and the implicit constraints specified by the search procedure. Examples of this type of learning system are described by Venkataraman (2001) and Batchelder (2002). The models underlying these systems are very similar; we describe only Venkataraman's work in detail.

Venkataraman proposes a method of word segmentation based on maximum-likelihood estimation. He discusses three different generative models of increasing complexity; we focus our analysis on the simplest of these, although our argument can be extended to all three. This model is a standard unigram model, i.e., it assumes that words are generated independently at random. The observed data consists of a corpus of phonemically transcribed child-directed speech, where utterance boundaries (corresponding to pauses in the input) are known, but word boundaries are unknown (see Figure 1). The probabilistic model underlying this system describes how to generate a corpus given U, the number of utterances in the corpus; the distinguished symbol \$, which is used to mark utterance boundaries; and Σ, the phonemic symbol alphabet (which does not include \$):

Repeat U times:

Repeat until \$ is generated:

1. Generate the next word, w, with probability $P_w(w)$.
2. Generate \$ with probability $p_\$$.

where P_w is some probability distribution over Σ^+, the set of all possible words. As each word is generated, it is concatenated onto the previously generated sequence of words. No boundary marker is added unless the end-of-utterance marker, \$, is generated. Under this model, the probability of generating words $w_1 \dots w_n$ as a single utterance is

$$\begin{aligned} P(w_1 \dots w_n\$) &= \left[\prod_{i=1}^{n-1} P_w(w_i)(1-p_\$)\right] P_w(w_n)p_\$ \\ &= \frac{p_\$}{1-p_\$} \prod_{i=1}^{n} P_w(w_i)(1-p_\$) \end{aligned} \quad (1)$$

and the probability of generating the unsegmented utterance u is found by summing over all possible sequences of words that could be concatenated to form u:

$$P(u) = \sum_{w_1 \dots w_n = u} P(w_1 \dots w_n\$)$$

The probability of the entire corpus is the product of the probabilities of the individual utterances. The hypothesis space for this model consists of all the possible assignments of probability values to words and the utterance boundary marker, i.e., possible values for $p_\$$ and the parameters of P_w.

Notice that in the hypothesis space just defined, some choices of P_w may assign non-zero probability to only a finite subset of potential words. Different hypotheses will have different sizes, i.e., different numbers of words will have non-zero probability. Crucially, however, no preference is given to hypotheses of any particular size — the maximum-likelihood assumption states that we should choose whichever hypothesis assigns the highest probability to the observed data.

What is the maximum-likelihood hypothesis under this model? It is straightforward to show that, in general, the maximum-likelihood solution for a model is the probability distribution that is closest to the empirical distribution (relative frequencies) of observations in the corpus, where the "distance" is computed by an information theoretic measure called the Kullback-Leibler divergence (Bishop, 2006, p. 57, inter al.) In the above model, there is one hypothesis that is able to match the empirical distribution of the corpus exactly. This hypothesis treats each

(a)	(b)
yuwanttusiD6bUk	you want to see the book
lUkD*z6b7wIThIzh&t	look there's a boy with his hat
&nd6dOgi	and a doggie
yuwanttulUk&tDIs	you want to look at this
lUk&tDIs	look at this
h&v6drINk	have a drink
okenQ	okay now
WAtsDIs	what's this
WAtsD&t	what's that
WAtIzIt	what is it
lUkk&nyutekItQt	look can you take it out
tekItQt	take it out
yuwantItIn	you want it in
pUtD&tan	put that on
D&t	that

Figure 1: An excerpt from the beginning of the corpus used as input to Venkataraman's (2001) word segmentation system, showing (a) the actual input corpus and (b) the corresponding standard orthographic transcription. The corpus was originally prepared by Brent (1996) using data from Bernstein-Ratner (1987), and was also used as input to Brent's (1999) MBDP-1 system.

utterance as a single "word", with probability equal to the empirical probability of that utterance in the corpus, and assumes $p_\$ = 1$. In other words, this solution memorizes the entire data set without segmenting utterances at all, and assigns zero probability to any unobserved utterance. Intuitively, any solution that does hypothesize word boundaries will require $p_\$ < 1$, which means that some unobserved utterances will have non-zero probability — those that can be created by, for example, concatenating two observed utterances, or rearranging the hypothesized words into novel orderings. Since some probability mass is allocated to these unobserved utterances, the probability of the observed data must be lower than in the case where $p_\$ = 1$ and no generalization is possible.

For a maximum-likelihood learner using the model in Equation 1, then, only a trivial segmentation will be found unless some constraint is placed on the kinds of hypotheses that are considered. Crucially, however, this argument does not depend on the particular form of P_w used in Equation 1, where words are assumed to be generated independent of context. Many other possible distributions over words would yield the same result. Venkataraman, for example, presents two other models in which the unigram distribution P_w is replaced with a *bigram* or *trigram* distribution: rather than generating words independent of context, each word is generated conditioned on either one or two previous words. That is, the bigram model defines

$$P(w_1 \ldots w_n\$) = P(w_1|\$) \left[\prod_{i=2}^{n} P(w_i|w_{i-1}) \right] P(\$|w_n).$$

Essentially, the reason that all of these models yield the same maximum-likelihood solution (an unsegmented corpus) is that they are allowed to consider hypotheses with arbitrary numbers of lexical items. When comparing hypotheses with different levels of complexity (corresponding here to the number of word types in the hypothesis), a maximum-likelihood learner will generally prefer a more complex hypothesis over a simple one. This leads to the problem of *overfitting*,

where the learner chooses a hypothesis that fits the observed data very well, but generalizes very poorly to new data. In the case of word segmentation, the solution of complete memorization allows the learner to fit the observed data perfectly. Since we know that this is not the solution found by Venkataraman's learners, we must conclude that the algorithm he proposes to search the space of possible hypotheses must be imposing additional constraints beyond those of the models themselves. It should be clear from the previous discussion that this is not the approach advocated here, since it renders constraints implicit and difficult to examine. Batchelder's (2002) maximum-likelihood learning system uses an explicit "external constraint" to penalize lexical items that are too long. This approach is a step in the right direction, but is less mathematically principled than Bayesian modeling, in which a (non-uniform) prior distribution over hypotheses is used within the model itself to constrain learning. We now review several of the Bayesian models that served as inspiration for our own work.

2.1.2 Bayesian models

In the previous section, we argued that unconstrained maximum-likelihood estimation is a poor way to choose between hypotheses with different complexities. In Bayesian modeling, the effect of the likelihood can be counterbalanced by choosing a prior distribution that favors simpler hypotheses. Simpler hypotheses will tend not to fit the observed data as well, but will tend to generalize more successfully to novel data. By considering both the likelihood and prior in determining the posterior probability of each hypothesis, Bayesian learners naturally avoid the kind of overfitting that maximum-likelihood learners encounter. The trade-off between fit and generalization will depend on exactly how the prior is defined; we now describe several methods that have been used to define priors in previous Bayesian models.

Perhaps the most well-known framework for defining Bayesian models is known as *minimum description length* (MDL) (Rissanen, 1989), and is exemplified by the work of de Marcken (1995) and Brent and Cartwright (1996). MDL is a particular formulation of Bayesian learning that has been used successfully in a number of other areas of language acquisition as well (Ellison, 1994; Goldsmith, 2001; Goldwater and Johnson, 2004; Creutz and Lagus, 2002; Dowman, 2000). The basic idea behind MDL is to define some encoding scheme that can be used to encode the corpus into a more compact representation. In word segmentation, for example, a code might consist of a list of lexical items along with a binary representation for each one. With appropriate choices for the lexical items and binary representations (with shorter representations assigned to more common words), the length of the corpus could be reduced by replacing each word with its binary code. In this framework, the learner's hypotheses are different possible encoding schemes. The minimum description length principle states that the optimal hypothesis is the one that minimizes the *combined length*, in bits, of the hypothesis itself (the codebook) and the encoded corpus. Using results from information theory, it can be shown that choosing a hypothesis using the MDL principle is equivalent to choosing the maximum *a posteriori* (MAP) hypothesis – the hypothesis with the highest posterior probability – under a Bayesian model where the prior probability of a hypothesis decreases exponentially with its length. In other words, MDL corresponds to a particular choice of prior distribution over hypotheses, where hypotheses are preferred if they can be described more succinctly.

Although MDL models can in principle produce good word segmentation results, there are no standard search algorithms for these kinds of models, and it is often difficult to design efficient model-specific algorithms. For example, Brent and Cartwright (1996) were forced to limit their analysis to a very short corpus (about 170 utterances) due to efficiency concerns. In later research, Brent developed another Bayesian model for word segmentation with a more efficient search algorithm (Brent, 1999). He named this system Model-Based Dynamic Programming (MBDP-1).[3] Since we will be returning to this model at various points throughout this paper,

[3]The 1 in MBDP-1 was intended as a version number, although Brent never developed any later versions of the

we now describe MBDP-1 in more detail.

Unlike models developed within the MDL framework, where hypotheses correspond to possible encoding methods, MBDP-1 assumes that the hypotheses under consideration are actual sequences of words, where each word is a sequence of phonemic symbols. The input corpus consists of phonemically transcribed utterances of child-directed speech, as in Figure 1. Some word sequences, when concatenated together to remove word boundaries, will form exactly the string of symbols found in the corpus, while others will not. The probability of the observed data given a particular hypothesized sequence of words will therefore either be equal to 1 (if the concatenated words form the corpus) or 0 (if not). Consequently, only hypotheses that are consistent with the corpus must be considered. For each possible segmentation of the corpus, the posterior probability of that segmentation will be directly proportional to its prior probability. The prior probability, in turn, is computed using a generative model. This model assumes that the sequence of words in the corpus was created in a sequence of four steps:[4]

Step 1 Generate the number of types that will be in the lexicon.

Step 2 Generate a token frequency for each lexical type.

Step 3 Generate the phonemic representation of each type (except for the single distinguished "utterance boundary" type, $).

Step 4 Generate an ordering for the set of tokens.

Each step in this process is associated with a probability distribution over the possible outcomes of that step, so together these four steps define the prior probability distribution over all possible segmented corpora. We discuss the specific distributions used in each step in Appendix B; here it is sufficient to note that these distributions tend to assign higher probability to segmentations containing fewer and shorter lexical items, so that the learner will prefer to split utterances into words.

To search the space of possible segmentations of the corpus, Brent develops an efficient *online* algorithm. The algorithm makes a single pass through the corpus, segmenting one utterance at a time based on the segmentations found for all previous utterances. The online nature of this algorithm is intended to provide a more realistic simulation of human word segmentation than earlier *batch* learning algorithms (de Marcken, 1995; Brent and Cartwright, 1996), which assume that the entire corpus of data is available to the learner at once (i.e., the learner may iterate over the data many times).

In the remainder of this paper, we will describe two new Bayesian models of word segmentation inspired, in part, by Brent's work. Like Brent, we use a generative model-based Bayesian framework to develop our learners. Moreover, as we prove in Appendix B, our first (unigram) model is mathematically very similar to the MBDP-1 model. However, our work differs from Brent's in two respects. First, our models are more flexible, which allows us to more easily investigate the effects of different modeling assumptions. In theory, each step of Brent's model can be individually modified, but in practice the mathematical statement of the model and the approximations necessary for the search procedure make it difficult to modify the model in any interesting way. In particular, the fourth step assumes a uniform distribution over orderings, which creates a unigram constraint that cannot easily be changed. We do not suppose that Brent was theoretically motivated in his choice of a unigram model, or that he would be opposed to introducing word-to-word dependencies, merely that the modeling choices available to him were limited by the statistical techniques available at the time of his work. In this paper, we make use of more flexible recent techniques that allows us to develop both unigram and bigram models of word segmentation and explore the differences in learning that result.

system.

[4]Our presentation involves a small change from Brent (1999), switching the order of Steps 2 and 3. This change makes no difference to the model, but provides a more natural grouping of steps for purposes of our analysis in Appendix B.

The second key contribution of this paper lies in our focus on analyzing the problem of word segmentation at the computational level by ensuring, to the best of our ability, that the only constraints on the learner are those imposed by the model itself. We have already shown that the model-based approaches of Venkataraman (2001) and Batchelder (2002) are constrained by their choice of search algorithms; in the following section we demonstrate that the approximate search procedure used by Brent (1999) prevents his learner, too, from identifying the optimal solution under his model. Although in principle one could develop a Bayesian model within the MDL or MBDP frameworks that could account for word-to-word dependencies, the associated search procedures would undoubtedly be even more complex than those required for the current unigram models, and thus even less likely to identify optimal solutions. Because our own work is based on more recent Bayesian techniques, we are able to develop search procedures using a standard class of algorithms known as Markov chain Monte Carlo methods (Gilks et al., 1996), which produce samples from the posterior distribution over hypotheses. We provide evidence that the solutions identified by our algorithms are indeed optimal or near-optimal, which allows us to draw conclusions using ideal observer arguments and to avoid the obfuscating effects of ad hoc search procedures.

3 Unigram model

3.1 Generative model

Like MBDP-1, our models assume that the hypotheses under consideration by the learner are possible segmentations of the corpus into sequences of words. Word sequences that are consistent with the corpus have a likelihood of 1, while others have a likelihood of 0, so the posterior probability of a segmentation is determined by its prior probability. Also as in MBDP-1, we compute the prior probability of a segmentation by assuming that the sequence of words in the segmentation was created according to a particular probabilistic generative process. Let $\mathbf{w} = w_1 \dots w_N$ be the words in the segmentation. Setting aside the complicating factor of utterance boundaries, our unigram model assumes that the ith word in the sequence, w_i, is generated as follows:

(1) Decide if w_i is a novel lexical item.

(2) a. If so, generate a phonemic form (phonemes $x_1 \dots x_M$) for w_i.
b. If not, choose an existing lexical form ℓ for w_i.

We assign probabilities to each possible choice as follows:

(1) $P(w_i \text{ is novel}) = \frac{\alpha_0}{n+\alpha_0}$, $P(w_i \text{ is not novel}) = \frac{n}{n+\alpha_0}$

(2) a. $P(w_i = x_1 \dots x_M \,|\, w_i \text{ is novel}) = p_\#(1-p_\#)^{M-1} \prod_{j=1}^{M} P(x_j)$
b. $P(w_i = \ell \,|\, w_i \text{ is not novel}) = \frac{n_\ell}{n}$

where α_0 is a parameter of the model, n is the number of previously generated words ($= i-1$), n_ℓ is the number of times lexical item ℓ has occurred in those n words, and $p_\#$ is the probability of generating a word boundary. Taken together, these definitions yield the following distribution over w_i given the previous words $\mathbf{w}_{-i} = \{w_1 \dots w_{i-1}\}$:

$$P(w_i = \ell \,|\, \mathbf{w}_{-i}) = \frac{n_\ell}{i-1+\alpha_0} + \frac{\alpha_0 P_0(w_i = \ell)}{i-1+\alpha_0} \tag{2}$$

where we use P_0 to refer to the unigram phoneme distribution in Step 2a. (The $p_\#$ and $1-p_\#$ factors in this distribution result from the process used to generate a word from constituent

phonemes: after each phoneme is generated, a word boundary is generated with probability $p_\#$ and the process ends, or else no word boundary is generated with probability $1-p_\#$ and another phoneme is generated.)

We now provide some intuition for the assumptions that are built into this model. First, notice that in Step 1, when n is small, the probability of generating a novel lexical item is relatively large. As more word tokens are generated and n increases, the relative probability of generating a novel item decreases, but never disappears entirely. This part of the model means that segmentations with too many different lexical items will have low probability, providing pressure for the learner to identify a segmentation consisting of relatively few lexical items. In Step 2a, we define the probability of a novel lexical item as the product of the probabilities of each of its phonemes. This ensures that very long lexical items will be strongly dispreferred. Finally, in Step 2b, we say that the probability of generating an instance of the lexical item ℓ is proportional to the number of times ℓ has already occurred. In effect, the learner assumes that a few lexical items will tend to occur very frequently, while most will occur only once or twice. In particular, the distribution over word frequencies produced by our model becomes a power-law distribution for large corpora (Arratia et al., 1992), the kind of distribution that is found in natural language (Zipf, 1932).

The model we have just described is an instance of a kind of model known in the statistical literature as a *Dirichlet process* (Ferguson, 1973). The Dirichlet process is commonly used in Bayesian statistics as a nonparametric prior for clustering models, and is closely related to Anderson's (1991) rational model of categorization (Sanborn et al., 2006b). The Dirichlet process has two parameters: the *concentration parameter* α_0 and the *base distribution* P_0. The concentration parameter determines how many clusters will typically be found in a data set of a particular size (here, how many word types for a particular number of tokens), and the base distribution determines the typical characteristics of a cluster (here, the particular phonemes in a word type). A more detailed mathematical treatment of our model and its relationship to the Dirichlet process is provided in Appendix A, but this connection leads us to refer to our unigram model of word segmentation as the "Dirichlet process" (DP) model.

So far, the model we have described assigns probabilities to sequences of words where there are no utterance boundaries. However, because the input corpus contains utterance boundaries, we need to extend the model to account for them. In the extended model, each hypothesis consists of a sequence of words and utterance boundaries, and hypotheses are consistent with the input if removing word boundaries (but not utterance boundaries) yields the input corpus. To compute the probability of a sequence of words and utterance boundaries, we assume that this sequence was generated using the model above, with the addition of an extra step: after each word is generated, an utterance boundary marker \$ is generated with probability $p_\$$ (or not, with probability $1 - p_\$$). For simplicity, we will suppress this portion of the model in the main body of this paper, and refer the reader to Appendix A for full details.

3.2 Inference

We have now defined a generative model that allows us to compute the probability of any segmentation of the input corpus. We are left with the problem of *inference*, or actually identifying the highest probability segmentation from among all possibilities. We used a method known as *Gibbs sampling* (Geman and Geman, 1984), a type of Markov chain Monte Carlo algorithm (Gilks et al., 1996) in which variables are repeatedly sampled from their conditional posterior distribution given the current values of all other variables in the model. Gibbs sampling is an iterative procedure in which (after a number of iterations used as a "burn-in" period to allow the sampler to converge) each successive iteration produces a sample from the full posterior distribution $P(h|d)$. In our sampler, the variables of interest are potential word boundaries, each of which can take on two possible values, corresponding to a word boundary or no word boundary.

Boundaries may be initialized at random or using any other method; initialization does not matter since the sampler will eventually converge to sampling from the posterior distribution.[5] Each iteration of the sampler consists of stepping through every possible boundary location and resampling its value conditioned on all other current boundary placements. Since each set of assignments to the boundary variables uniquely determines a segmentation, sampling boundaries is equivalent to sampling sequences of words as our hypotheses. Although Gibbs sampling is a *batch* learning algorithm, where the entire data set is available to the learner at once, we note that there are other sampling techniques known as *particle filters* (Doucet et al., 2000; Sanborn et al., 2006b) that can be used to produce approximations of the posterior distribution in an *online* fashion (examining each utterance in turn exactly once). We return in the General Discussion to the question of how a particle filter might be developed for our own model in the future. Full details of our Gibbs sampling algorithm are provided in Appendix A.

3.3 Simulations

3.3.1 Data

To facilitate comparison to previous models of word segmentation, we report results on the same corpus used by Brent (1999) and Venkataraman (2001). The data is derived from the Bernstein-Ratner corpus (Bernstein-Ratner, 1987) of the CHILDES database (MacWhinney and Snow, 1985), which contains orthographic transcriptions of utterances directed at 13- to 23-month-olds. The data was post-processed by Brent, who removed disfluencies and non-words, discarded parental utterances not directed at the children, and converted the rest of the words into a phonemic representation using a phonemic dictionary (i.e. each orthographic form was always given the same phonemic form). The resulting corpus contains 9790 utterances, with 33399 word tokens and 1321 unique types. The average number of words per utterance is 3.41 and the average word length (in phonemes) is 2.87. The word boundaries in the corpus are used as the gold standard for evaluation, but are not provided in the input to the system (except for word boundaries that are also utterance boundaries).

The process used to create this corpus means that it is missing many of the complexities of real child-directed speech. Not the least of these is the acoustic variability with which different tokens of the same word are produced, a factor which presumably makes word segmentation more difficult. On the other hand, the corpus is also missing many cues which could aid in segmentation, such as coarticulation information, stress, and duration. While this idealization of child-directed speech is somewhat unrealistic, the corpus does provide a way to investigate the use of purely distributional cues for segmentation, and permits direct comparison to other word segmentation systems.

3.3.2 Evaluation procedure

For quantitative evaluation, we adopt the same measures used by Brent (1999) and Venkataraman (2001): precision (number of correct items found out of all items found) and recall (number of correct items found out of all correct items). These measures are widespread in the computational linguistics community; the same measures are often known as *accuracy* and *completeness* in the cognitive science community (Brent and Cartwright, 1996; Christiansen et al., 1998). We also report results in terms of F_0 (another common metric used in computational linguistics, also known as F-measure or F-score). F_0 is the geometric average of precision and recall, defined as

[5] Of course, our point that initialization does not matter is a theoretical one; in practice, some initializations may lead to faster convergence than others, and checking that different initializations lead to the same results is one way of testing for convergence of the sampler, as we do in Appendix A.

$\frac{2*\text{precision}*\text{recall}}{\text{precision}+\text{recall}}$, and penalizes results where precision and recall are very different. We report the following scores for each model we propose:

- P, R, F: precision, recall, and F_0 on words: both boundaries must be correctly identified to count as correct.
- LP, LR, LF: precision, recall, and F_0 on the lexicon, i.e. word types.
- BP, BR, BF: precision, recall, and F_0 on potentially ambiguous boundaries (i.e. utterance boundaries are not included in the counts).

As an example, imagine a one-utterance corpus whose correct segmentation is `look at the big dog there`, where instead we find the segmentation `look at the bigdo g the re`. There are seven words in the found segmentation, and six in the true segmentation; three of these words match. We report all scores as percentages, so P = 42.9% (3/7), R= 50.0% (3/6), and F=46.2%. Similarly, BP = 66.7% (4/6), BR = 80.0% (4/5), BF = 72.7%, LP= 50.0% (3/6), LR= 50.0% (3/6), and LF= 50.0%. Note that if the learner correctly identifies all of the boundaries in the true solution, but also proposes extra boundaries (*oversegmentation*), then boundary recall will reach 100%, but boundary precision and boundary F_0 will be lower. Conversely, if the learner proposes no incorrect boundaries, but fails to identify all of the true boundaries (*undersegmentation*), then boundary precision will be 100%, but boundary recall and F_0 will be lower. In either case, scores for word tokens and lexical items will be below 100%.

For comparison, we report scores as well for Brent's MBDP-1 system (Brent, 1999) and Venkataraman's n-gram segmentation systems (Venkataraman, 2001), which we will refer to as NGS-u and NGS-b (for the unigram and bigram models). Both Brent and Venkataraman use online search procedures (i.e., their systems make a single pass through the data, segmenting each utterance in turn), so in their papers they calculate precision and recall separately on each 500-utterance block of the corpus and graph the results to show how scores change as more data is processed. They do not report lexicon recall or boundary precision and recall. Their results are rather noisy, but performance seems to stabilize rapidly, after about 1500 utterances. To facilitate comparison with our own results, we calculated scores for MBDP-1 and NGS over the whole corpus, using Venkataraman's implementations of these algorithms.[6]

Since our algorithm produces random segmentations sampled from the posterior distribution rather than a single optimal solution, there are several possible ways to evaluate its performance. For most of our simulations, we evaluated a single sample taken after 20,000 iterations. We used a method known as *simulated annealing* (Aarts and Korst, 1989) to speed convergence of the sampler, and in some cases (noted below) to obtain an approximation of the MAP solution by concentrating samples around the mode of the posterior. This allowed us to examine possible differences between a random sample of the posterior and a sample more closely approximating the MAP segmentation. Details of the annealing and MAP approximation procedures can be found in Appendix A.

3.3.3 Results and Discussion

The DP model we have described has two free parameters: $p_\#$ (the prior probability of a word boundary), and α_0 (which affects the number of word types proposed).[7] Figure 2 shows the effects of varying of $p_\#$ and α_0. Lower values of $p_\#$ result in more long words, which tends to improve recall (and thus F_0) in the lexicon. The accompanying decrease in token accuracy is due to an increasing tendency for the model to concatenate short words together, a phenomenon

[6]The implementations are available at http://www.speech.sri.com/people/anand/.

[7]The DP model actually contains a third free parameter, ρ, used as a prior over the probability of an utterance boundary (see Appendix A). Given the large number of known utterance boundaries, the value of ρ should have little effect on results, so we simply fixed $\rho = 2$ for all simulations.

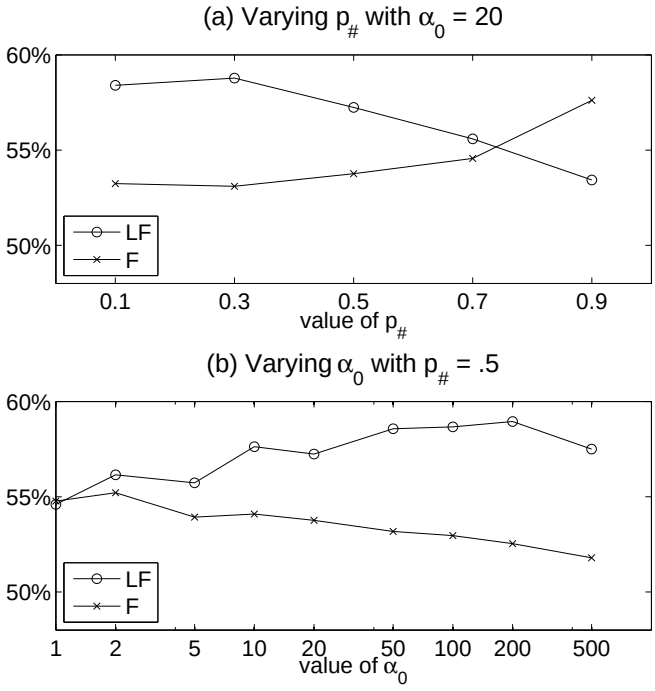

Figure 2: F_0 for words (F) and lexical items (LF) in the DP model (a) as a function of $p_\#$, with $\alpha_0 = 20$ and (b) as a function of α_0, with $p_\# = .5$.

we discuss further below. Higher values of α_0 allow more novel words, which also improves lexicon recall, but begins to degrade precision after a point. Due to the negative correlation between token accuracy and lexicon accuracy, there is no single best value for either $p_\#$ or α_0. In the remainder of this section, we focus on the results for $p_\# = .5, \alpha_0 = 20$ (though others are qualitatively similar – we discuss these briefly below).

In Table 1, we compare the results of our system to those of MBDP-1 and NGS-u. Although our system has higher lexicon accuracy than the others, its token accuracy is much worse. Performance does not vary a great deal between different samples, since calculating the score for a single sample already involves averaging over many random choices – the choices of whether to place a boundary at each location or not. Table 2 shows the mean and standard deviation in F_0 scores and posterior probabilities over samples taken from 10 independent runs of the algorithm with different random initializations. The same statistics are also provided for ten samples obtained from a single run of the sampler. Samples from a single run are not independent, so to reduce the amount of correlation between these samples they were taken at 100-iteration intervals (at iterations 19100, 19200, ... 20000). Nevertheless, they show less variability than the truly independent samples. In both cases, lexicon accuracy is more variable than token accuracy, probably because there are far fewer lexical items to average over within a single sample. Finally, Table 2 provides results for the approximate MAP evaluation procedure. This procedure is clearly imperfect, since if it were able to identify the true MAP solution, there would be no difference in results across multiple runs of the algorithm. In fact, compared to the standard sampling procedure, there is only slightly less variation in F_0 scores, and greater variation

Table 1: Word segmentation accuracy of unigram systems

	Performance measure								
Model	P	R	F	BP	BR	BF	LP	LR	LF
NGS-u	**67.7**	**70.2**	**68.9**	80.6	**84.8**	**82.6**	52.9	51.3	52.0
MBDP-1	67.0	69.4	68.2	80.3	84.3	82.3	53.6	51.3	52.4
DP	61.9	47.6	53.8	**92.4**	62.2	74.3	**57.0**	**57.5**	**57.2**

Note: P, R, and F are precision, recall, and F_0 for word tokens; BP, LP, etc. are the corresponding scores for ambiguous boundaries and lexical items. Best scores are shown in bold. DP results are with $p_{\#} = .5$ and $\alpha_0 = 20$.

Table 2: Results of the DP model, averaged over multiple samples

	F	LF	$-\log P(\mathbf{w})$
Samples (10 runs)	53.9 (.32)	57.8 (.60)	200587 (192)
Samples (1 run)	53.5 (.07)	57.7 (.43)	200493 (25)
MAP approx.	53.7 (.26)	58.7 (.56)	199853 (228)

Note: Token F_0 (F), lexicon F_0 (LF), and negative log posterior probability were averaged over 10 samples from independent runs of our Gibbs sampler, over 10 samples from a single run, and over 10 samples from independent runs of our MAP approximation (see text). Standard deviations are shown in parentheses.

in probability.[8] Nevertheless, the MAP approximation does succeed in finding solutions with significantly higher probabilities. These solutions also have higher lexicon accuracy, although token accuracy remains low.

The reason that token accuracy is so low with the DP model is that it often mis-analyzes frequently occurring words. Many instances of these words occur in common collocations such as *what's that* and *do you*, which the system interprets as a single words. This pattern of errors is apparent in the boundary scores: boundary precision is very high, indicating that when the system proposes a boundary, it is almost always correct. Boundary recall is low, indicating undersegmentation.

We analyzed the behavior of the system more carefully by examining the segmented corpus and lexicon. A full 30% of the proposed lexicon and nearly 30% of tokens consist of undersegmentation (collocation) errors, while only 12% of types and 5% of tokens are other non-words. (Some additional token errors, under 4%, are caused by proposing a correct word in an incorrect location.) About 85% of collocations (both types and tokens) are composed of two words, nearly all the rest are three words. To illustrate the phenomenon, we provide the system's segmentation of the first 40 utterances in the corpus in Figure 3, and the 35 most frequently found lexical items in Figure 4. The 70 most frequent collocations identified as single words by the system are shown in Figure 5.

It is interesting to examine the collocations listed in Figure 5 with reference to the existing literature on children's early representation of words. Peters (1983), in particular, provides a number of examples of children's undersegmentation errors (using their productions as evidence).

[8]The large standard deviation in the probabilities of the approximate MAP solutions is due to a single outlier. The standard deviation among the remaining nine solutions is 160, well below the standard deviation in the sample solutions, where there are no outliers.

```
youwant to see thebook              let'ssee
look there's aboy with his hat      yeah
and adoggie                         pull itout
you wantto lookatthis               what's it
lookatthis                          look
havea drink                         look
okay now                            what'sthat
what'sthis                          get it
what'sthat                          getit
whatisit                            getit
look canyou take itout              isthat for thedoggie
take itout                          canyou feed it to thedoggie
youwant it in                       feed it
put that on                         putit in okay
that                                okay
yes                                 whatareyou gonna do
okay                                I'll let her playwith this fora while
open itup                           what
take thedoggie out                  what
ithink it will comeout              what'sthis
```

Figure 3: The first 40 utterances in the corpus as segmented by the DP model (represented orthographically for readability), illustrating that the model undersegments the corpus. The stochastic nature of the Gibbs sampling procedure is apparent: some sequences, such as `youwantto` and `getit`, receive two different analyses.

+	396	what	1704 you
+	383	you	1291 the
+	360	okay	895 a
+	351	to/too/two	798 that
+	340	and	783 what
+	337	yeah	653 is
+	313	the	632 it
+	295	it	588 this
+	293	look	569 what’s
-	275	[z]	528 to/too/two
-	258	what’sthat	463 do
+	248	that	429 look
+	235	a	412 can
+	217	no	399 that’s
+	196	that’s	389 see
+	193	there/their	389 there/their
+	189	this	382 I
+	188	see	378 and
-	184	canyou	375 in
+	178	in	363 your/you’re
+	177	your/you’re	362 are
+	172	here	360 okay
+	160	what’s	337 yeah
+	147	it’s	301 no
-	141	ing	268 like
-	138	isthat	266 it’s
-	135	what’sthis	250 on
+	123	is	246 here
+	117	do	246 one
+	116	now	244 want
-	112	[s]	239 put
+	104	for	227 he
-	102	thedoggie	226 wanna
-	101	that’sa	221 right
+	101	book	217 book

Figure 4: The 35 most frequent items in the lexicon found by the DP model (left) and in the correct lexicon (right). Except for the phonemes **z** and **s**, lexical items are represented orthographically for readability. Different possible spellings of a single phonemic form are separated by slashes. The frequency of each lexical item is shown to its left. Items in the segmented lexicon are indicated as correct (+) or incorrect (-). Frequencies of correct items in the segmented lexicon are lower than in the true lexicon because many occurrences of these items are accounted for by collocations.

Full S or socialization:

258 what's that
135 what's this
65 thank you
61 that's right
57 bye bye
53 what is it
41 look at this
38 what are those
38 what else
36 who's that
33 that one
31 night night
30 let me out
30 sit down
30 close the door
29 good girl
28 look at that

Wh + X:

44 where's the
39 how many
34 what are you
30 what do you

Det + N:

84 the doggie
57 the dragon
47 this one
42 the book
39 the dog
35 the boy
35 the bunny
36 this book
31 a book
30 the door
29 your hand
29 another one
29 that one

Aux + NP (+ V):

183 can you
138 is that
91 do you
56 do you want
53 would you like
50 is it
48 did you
39 do you see
30 are you
29 is he

Pronoun + Aux/V:

94 you want
84 you can
65 you wanna
55 I think
48 you like
34 you're gonna
33 you don't
31 I see
28 you know what
28 I don't

Other:

100 that's a
87 look at
75 it's a
69 in there
51 this is
48 on the
42 those are
40 is for
39 put it
38 do it
36 see the
36 in the
32 play with
30 put him
28 kind of
27 wanna see

Figure 5: The 70 most frequently occurring items in the segmented lexicon that consist of multiple words from the true lexicon. These items are all identified as single words; the true word boundaries have been inserted for readability. The frequency of each item is shown to its left.

Several of the full sentences and social conventions in Figure 5 (e.g., *thank you*, *that's right*, *bye bye*, *look at this*) are included among her examples. In addition, some of the other collocation errors found by our unigram system match the examples of "formulaic frames" given by Peters: the "verb introducer" *can you* and "noun introducers" *it's a*, *this is*, *those are*, and *see the*. Phonological reductions in adult speech also suggest that a few of the collocations found by the system (e.g., *did you*, *what do you*) may even be treated as single units by adults in some circumstances. However, the extent and variety of collocations found by the system is certainly much broader than what researchers have so far found evidence for in young children.

We will defer for the moment any further discussion of whether children's early word representations are similar to those found by our DP model (we return to this issue in the General Discussion), and instead turn to the question of why these units are found. The answer seems clear: groups of words that frequently co-occur violate the unigram assumption in the model, since they exhibit strong word-to-word dependencies. The only way the learner can capture these dependencies is by assuming that these collocations are in fact words themselves. As an example, consider the word *that*. In our corpus, the empirical probability of the word *that* is $798/33399 \approx .024$. However, the empirical probability of *that* following the word *what's* is far higher: $263/569 \approx .46$. Since the strong correlation between *what's* and *that* violates the independence assumption of the model, the learner concludes that *what'sthat* must be a single word.

Note that by changing the values of the parameters α_0 and $p_\#$, it is possible to reduce the level of undersegmentation, but only slightly, and at the cost of introducing other errors. For example, raising the value of $p_\#$ to 0.9 strongly increases the model's preference for short lexical items, but collocations still make up 24% of both types and tokens in this case. Measures of token accuracy increase by a few points, but are still well below those of previous systems. The main qualitative difference between results with $p_\# = .5$ and $p_\# = .9$ is that with the higher value, infrequent words are more likely to be oversegmented into very short one- or two-phoneme chunks (reflected in a drop in lexicon accuracy). However, frequent words still tend to be undersegmented as before.

It is also worth noting that, although the proportion of collocations in the lexicons found by MBDP-1 and NGS-u is comparable to the proportion found by our own model (24%), only 6% of tokens found by these systems are collocation errors. This fact seems to contradict our analysis of the failures of our own unigram model, and raises a number of questions. Why don't these other unigram models exhibit the same problems as our own? Is there some other weakness in our model that might be causing or compounding the problems with undersegmentation? Is it possible to design a successful unigram model for word segmentation? We address these questions in the following section.

4 Other unigram models

4.1 MBDP-1 and search

In the previous section, we showed that the optimal segmentation under our unigram model is one that identifies common collocations as individual words. Our earlier discussion of Venkataraman's (2001) NGS models demonstrated that the optimal solution under those models is a completely unsegmented corpus. What about Brent's (1999) MBDP-1 model? While the definition of this unigram model makes it difficult to determine what the optimal solution is, our main concern was whether it exhibits the same problems with undersegmentation as our own unigram model. The results presented by Brent do not indicate undersegmentation, but it turns out that these results, like Venkataraman's, are influenced by the approximate search procedure used. We determined this by calculating the probability of various segmentations of the corpus under each model, as shown in Table 3. The results indicate that the MBDP-1 model assigns higher

Table 3: Negative log probabilities of various segmentations under each unigram model

	Segmentation				
Model	True	None	MBDP-1	NGS-u	DP
NGS-u	204.5	**90.9**	210.7	210.8	183.0
MBDP-1	208.2	321.7	217.0	218.0	**189.8**
DP	222.4	393.6	231.2	231.6	**200.6**

Note: Row headings identify the models used to evaluate each segmentation. Column headings identify the different segmentations: the true segmentation, the segmentation with no utterance-internal boundaries, and the segmentation found by each system. Actual log probabilities are 1000x those shown.

Table 4: Accuracy of the various systems on the permuted corpus

	Performance measure								
Model	P	R	F	BP	BR	BF	LP	LR	LF
NGS-u	76.6	85.8	81.0	83.5	97.6	90.0	60.0	52.4	55.9
MBDP-1	77.0	86.1	81.3	83.7	97.7	90.2	60.8	53.0	56.6
DP	**94.2**	**97.1**	**95.6**	**95.7**	**99.8**	**97.7**	**86.5**	**62.2**	**72.4**

Note: P, R, and F are precision, recall, and F_0 for word tokens; BP, LP, etc. are the corresponding scores for ambiguous boundaries and lexical items. Best scores are shown in bold. DP results are with $p_{\#} = .5$ and $\alpha_0 = 20$.

probability to the solution found by our Gibbs sampler than to the solution found by Brent's own incremental search algorithm. In other words, the model underlying MBDP-1 *does* favor the lower-accuracy collocation solution, but Brent's approximate search algorithm finds a different solution that has higher accuracy but lower probability under the model.

We performed two simulations suggesting that our own inference procedure does not suffer from similar problems. First, we initialized the Gibbs sampler in three different ways: with no utterance-internal boundaries, with a boundary after every character, and with random boundaries. The results were virtually the same regardless of initialization (see Appendix A for details). Second, we created an artificial corpus by randomly permuting all the words in the true corpus and arranging them into utterances with the same number of words as in the true corpus. This manipulation creates a corpus where the unigram assumption is correct. If our inference procedure works properly, the unigram system should be able to correctly identify the words in the permuted corpus. This is exactly what we found, as shown in Table 4. The performance of the DP model jumps dramatically, and most errors occur on infrequent words (as evidenced by the fact that token accuracy is much higher than lexicon accuracy). In contrast, MBDP-1 and NGS-u receive a much smaller benefit from the permuted corpus, again indicating the influence of search.

These results imply that the DP model itself, rather than the Gibbs sampling procedure we used for inference, is responsible for the poor segmentation performance on the natural language corpus. In particular, the unigram assumption of the model seems to be at fault. In the following section we present some additional simulations designed to further test this hypothesis. In these simulations, we change the model of lexical items used in Step 2a of the model, which has so far assumed that lexical items are created by choosing phonemes independently at random. If the

original poor lexical model is responsible for the DP model's undersegmentation of the corpus, then improving the lexical model should improve performance. However, if the problem is that the unigram assumption fails to account for sequential dependencies in the corpus, then a better lexical model will not make much difference.

4.2 The impact of the lexical model on word segmentation

One possible improvement to the lexical model is to replace the assumption of a uniform distribution over phonemes with the more realistic assumption that phonemes have different probabilities of occurrence. This assumption is more in line with the MBDP-1 and NGS models. In NGS, phoneme probabilities are estimated online according to their empirical distribution in the corpus. In MBDP-1, phoneme probabilities are also estimated online, but according to their empirical distribution in the current lexicon. For models like MBDP-1 and the DP model, where the phoneme distribution is used to generate lexicon items rather than word tokens, the latter approach makes more sense. It is relatively straightforward to extend the DP model to infer the phoneme distribution in the lexicon simultaneously with inferring the lexicon itself. Before implementing this extension, however, we tried simply fixing the phoneme distribution to the empirical distribution in the true lexicon. This procedure gives an upper bound on the performance that could be expected if the distribution were learned. We found that this change improved lexicon F_0 somewhat (to 60.5, with $\alpha = 20$ and $p_\# = .5$), but made almost no difference on token F_0 (53.6). Inference of the phoneme distribution was therefore not implemented.

Other changes could be made to the lexical model in order to create a better model of word shapes. For example, using a bigram or trigram phoneme model would allow the learner to acquire some notion of phonotactics. Basing the model on syllables rather than phonemes could incorporate constraints on the presence of vowels or syllable weight. Rather than testing all these different possibilities, we designed a simulation to determine an approximate upper bound on performance in the unigram DP model. In this simulation, we provided the model with information that no infant would actually have access to: the set of word types that occur in the correctly segmented corpus. The lexical model is defined as follows:

$$P_{true}(w_i = \ell) \quad = \quad \begin{cases} (1-\epsilon)\frac{1}{|L|} + \epsilon P_0(w_i = \ell) & \ell \in L \\ \epsilon P_0(w_i = \ell) & \ell \notin L \end{cases}$$

where L is the true set of lexical items in the data, and ϵ is some small mixing constant. In other words, this model is a mixture between a uniform distribution over the true lexical items and the basic model P_0. If $\epsilon = 0$, the model is constrained so that segmentations may only contain words from the true lexicon. If $\epsilon > 0$, a small amount of noise is introduced so that new lexical items are possible, but have much lower probability than the true lexical items. If the model still postulates collocations when ϵ is very small, we have evidence that the unigram assumption, rather than any failure in the lexicon model, is responsible for the problem.

The results from this model are shown in Table 5. Not surprisingly, the lexicon F_0 scores in this model are very high, and there is a large improvement in token F_0 scores against previous models. However, considering the amount of information provided to the model, its scores are still surprisingly low, and collocations remain a problem, especially for frequent items.

Considering the case where $\epsilon = 10^{-6}$ yields some insight into the performance of these models with improved lexical models. The solution found, with a lexicon consisting of 13.1% collocations, has higher probability than the true solution. This is despite the fact that the most probable incorrect lexical items are about five orders of magnitude less probable than the true lexical items.[9] These incorrect lexical items are proposed despite their extremely low

[9]There are 1321 lexical items in the corpus, so under the lexical model, the probability of each of these is approximately 10^{-3}. There are 50 phonemes and $p_\# = .5$, so a single-character word has probability .01 under P_0.

Table 5: Results of the DP model using P_{true}

Mixing constant	Accuracy F	Accuracy LF	% Collocations tokens	% Collocations lexicon
$\epsilon = 10^{-2}$	60.5	81.7	27.7	21.6
$\epsilon = 10^{-3}$	62.7	83.4	25.8	19.1
$\epsilon = 10^{-4}$	64.5	84.8	24.6	16.9
$\epsilon = 10^{-5}$	65.5	85.3	23.7	15.5
$\epsilon = 10^{-6}$	68.2	85.6	21.4	13.1

Note: Shown, for each value of ϵ, is token F_0 (F), lexicon F_0 (LF), and the percentage of tokens and lexical items that are multiword collocations.

probability because only the first occurrence of each word is accounted for by the lexical model. Subsequent occurrences are accounted for by the part of the model that generates repeated words, where probabilities are proportional to the number of previous occurrences. Therefore, low-probability lexical items incur no penalty (beyond that of any other word) after the first occurrence. This is why the collocations remaining in the DP model using P_{true} are the highest-frequency collocations: over many occurrences, the probability mass gained by modeling these collocations as single words outweighs the mass lost in generating the first occurrence.

The results of this simulation suggest that the large number of collocations found by the unigram DP model are not due to the weakness of the lexical model. Regardless of how good the lexical model is, it will not be able to completely overcome the influence of the unigram assumption governing word tokens when modeling the full corpus. In order to reduce the number of collocations, it is necessary to account for sequential dependencies between words. Before showing how to do so, however, we first present theoretical results regarding the generality of our conclusions about unigram models.

4.3 MBDP-1, the DP model, and other unigram models

The probabilistic models used in MBDP-1 and our Dirichlet process model appeal to quite different generative processes. To generate a corpus using MBDP, the number of word types is sampled, then the token frequencies, then the forms of the words in the lexicon, and finally an ordering for the set of tokens. Using the DP model, the length of the corpus (number of word tokens) must be chosen, and then the sequence of words is generated, implicitly determining the number of word types and the lexicon. Although these two approaches to generating a corpus are very different, it is possible to show that, by varying the specific distributions assumed at each step of the MBDP-1 generative process, the two approaches can result in exactly the same distribution over word sequences. In Appendix B we show that by changing how the size of the lexicon and the token frequencies are chosen in Steps 1 and 2 of the MBDP model, we can produce distributions over words that are equivalent to the distribution given by the DP model when conditioned on the total number of words.

This formal correspondence between MBDP-1 and the DP model suggests that the two models might express similar preferences about segmentations of corpora. In particular, the generative processes behind the two models share two components – the distributions over the lexicon and ordering of tokens – and differ only in the way that word frequencies are assigned. We can see the consequences of these shared components by comparing the probabilities that MBDP-

Multiplying by the discount factor $\epsilon = 10^{-6}$ yields $P_{true} = 10^{-8}$ for one-character words not in the true lexicon. Longer incorrect words will have much lower probability.

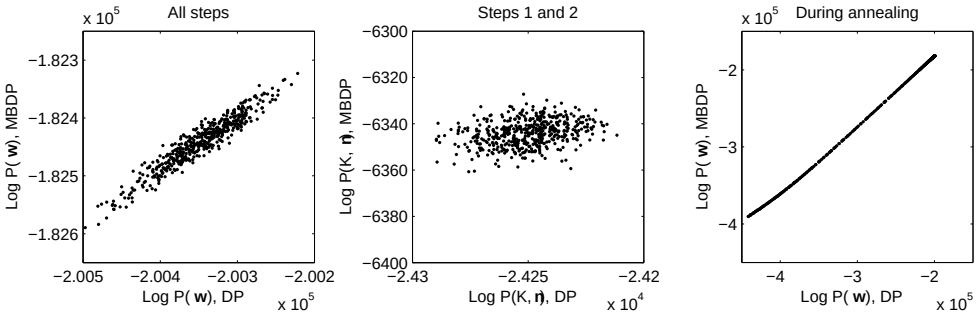

Figure 6: Left: Probabilities of 500 segmentations sampled from the posterior distribution of the DP model, computed under both that model and MBDP-1. Middle: Probabilities of the non-shared components of the same samples – the number of word types and their frequencies. Right: Probabilities under the two models of 500 segmentations obtained during annealing (to exhibit a wider range of quality).

1 and the DP model assign to different segmentations. We compared the probabilities that the models assigned to 500 samples of a full segmentation $\mathbf{w}$ taken from the last 1000 iterations of a 20,000-iteration simulation like the ones described in our unigram model simulations. As shown in Figure 6, these probabilities were highly correlated, with the linear correlation between the values of $\log P(\mathbf{w})$ under the two models being $r = .95$. This correlation is almost entirely due to the shared components of the generative processes: if we remove the shared factors corresponding to the probability of the word forms in the lexicon and the ordering of the tokens, the correlation is significantly reduced, becoming just $r = .30$.

This analysis indicates that MBDP-1 and the nonparametric Bayesian model show a very close correspondence in their preferences for different segmentations. The probability of a segmentation under the two models is highly correlated, despite the fact that they define quite different distributions over word frequencies. This effect is even more pronounced if we look at the relationship between the probabilities assigned by the two models to segmentations ranging in quality from very poor to very good. Using simulated annealing, we generated 500 segmentations of varying quality as defined by the DP model (details of the method are provided in Appendix A) and evaluated their probability under each model. As shown in Figure 6, the models agree strongly about the relative quality of those results. The dominant term in $\log P(\mathbf{w})$ for both models is that used in the fourth step: the distribution over orderings of word tokens.

The dominance of the distribution over the orderings of word tokens in determining the probability of the segmentation suggests a further conclusion. The distribution used in this step in both MBDP-1 and the DP model is uniform over all permutations of the word tokens in the corpus. Intuitively, this uniform distribution indicates that the order of the words does not matter, and expresses the unigram assumption that underlies these models. In Appendix B, we show that *any* unigram model has to make this assumption. Thus, all unigram models can be expected to be highly correlated with MBDP-1 and the DP model in their preferences regarding segmentations. This suggests that the problems that we have identified with the segmentations produced by the DP model generalize not just to MBDP-1, but also to other models that assume that words are independent units.

5 Bigram model

5.1 The hierarchical Dirichlet process model

In the previous section, we discussed empirical and theoretical evidence that defining words as statistically independent units leads to undersegmentation of natural language. We now ask whether modifying this assumption can lead to better segmentation. We address this question by developing a different model in which words are assumed to help predict other words. Recent work has suggested that children may be sensitive to statistical dependencies that range over several words (Mintz, 2002; Gómez and Maye, 2005). As a first step towards exploring the effects of such dependencies on word segmentation, we will define a model that considers only dependencies between adjacent words. This model assumes that the probability of a word depends on a single previous word of context, so the unit of dependency is a pair of words, or bigram.

Like our unigram model, our bigram model defines the probability of a segmentation by assuming that it was generated as a sequence of words $\mathbf{w} = w_1 \dots w_N$ using a probabilistic process. Unlike the unigram model, w_i is generated using a process that takes into account the previous (already generated) word in the sequence, w_{i-1}:

(1) Decide whether the pair $\langle w_{i-1}, w_i \rangle$ will be a novel bigram type.

(2) a. If so,
   i. Decide whether w_i will be a novel unigram type.
   ii. a. If so, generate a phonemic form (phonemes $x_1 \dots x_M$) for w_i.
   b. If not, choose an existing lexical form ℓ for w_i.
   b. If not, choose a lexical form ℓ for w_i from among those that have previously been generated following w_{i-1}.

Notice that Step 2a, which creates the second word of a novel bigram, uses the same steps we used in our unigram model. The unigram process in Step 2a generates a set of word types which the bigram process in Steps 1–2 assembles into bigrams.

The probabilities associated with the bigram generative process are

(1) $P(\langle w_{i-1}, w_i \rangle \text{ is a novel bigram} \,|\, w_{i-1} = \ell') = \frac{\alpha_1}{n_{\ell'} + \alpha_1}$

$P(\langle w_{i-1}, w_i \rangle \text{ is not a novel bigram} \,|\, w_{i-1} = \ell') = \frac{n_{\ell'}}{n_{\ell'} + \alpha_1}$

(2) a. i. $P(w_i \text{ is a novel word} \,|\, \langle w_{i-1}, w_i \rangle \text{ is a novel bigram}) = \frac{\alpha_0}{b + \alpha_0}$

$P(w_i \text{ is not a novel word} \,|\, \langle w_{i-1}, w_i \rangle \text{ is a novel bigram}) = \frac{b}{b + \alpha_0}$

ii. a. $P(w_i = x_1 \dots x_M \,|\, w_i \text{ is a novel word}) = P_0(x_1 \dots x_M)$

b. $P(w_i = \ell \,|\, w_i \text{ is not a novel word}) = \frac{b_\ell}{b}$

b. $P(w_i = \ell \,|\, \langle w_{i-1}, w_i \rangle \text{ is not a novel bigram and } w_{i-1} = \ell') = \frac{n_{\langle \ell', \ell \rangle}}{n_{\ell'}}$

where α_0 and α_1 are parameters of the model, P_0 is the lexical model defined as part of our unigram model, ℓ' is the lexical form of w_{i-1}, $n_{\ell'}$ and $n_{\langle \ell', \ell \rangle}$ are the number of occurrences in the first $i-1$ words of the unigram ℓ' and the bigram $\langle \ell', \ell \rangle$, b is the number of bigram types in the first $i-1$ words, and b_ℓ is the number of those types whose second word is ℓ.

The intuition behind this model is similar to that of the unigram model. Step 1 says that the more times ℓ' has been generated, the less likely a new word will be generated following it; this limits the number of bigram types. Step 2a is like the unigram generative process, except that the probabilities are defined in terms of bigram types instead of unigram tokens. The idea is that some words combine more promiscuously into bigrams than others: If ℓ has been generated in many different contexts already, it is more likely to be generated in this new context. Finally,

in Step 2b, the probability of generating ℓ following ℓ' is proportional to the number of times this pair has been generated already, which leads to a preference for power-law distributions over the second item in each bigram.

The bigram model we have just defined is known as a *hierarchical Dirichlet process* (HDP) (Teh et al., 2005). The HDP is an extension of the DP, and is typically used to model data in which there are multiple distributions over similar sets of outcomes, and the distributions are believed to be similar. For language modeling, we can define a bigram model by assuming that each word has a different distribution over the words that follow it, but all these distributions are linked by sharing a common set of unigrams. Again, we will use this formal connection to name the model, making our bigram model the "hierarchical Dirichlet process" or HDP model. Our HDP language model is similar to previously proposed n-gram models using hierarchical Pitman-Yor processes (Goldwater et al., 2006b; Teh, 2006). For further discussion and a more detailed presentation of the model, including a treatment of utterance boundaries, see Appendix A.

5.2 Simulations

5.2.1 Method

For our simulations with the bigram model, we used the same input corpus and evaluation measures as in our unigram model simulations. To identify a high-probability solution, we implemented a Gibbs sampler that is conceptually similar to the unigram sampler. Details can be found in Appendix A.[10] The sampler was initialized by assigning word boundaries at random in each utterance, although, as in the unigram model, other initialization methods yield results similar to those presented below. We experimented with various values for the three free parameters of the model, α_0, α_1, and $p_\#$.[11]

5.2.2 Results and discussion

Figure 7 plots the accuracy of our bigram model for various values of α_0, α_1, and $p_\#$ based on a single sample taken after 20,000 iterations. What we see from Figure 7 is that $p_\#$ (the probability of generating the word boundary marker when producing a novel word) has relatively little effect on results: lexicon accuracy is slightly higher for the lower value of $p_\#$, but segmentation accuracy is the same. α_0 (which determines the probability of generating a novel word) also primarily affects lexicon accuracy. Since higher values of α_0 lead to more novel words (i.e., lexical items), lexicon recall increases for higher values of α_0. Lexicon precision drops slightly, but the overall effect is that F_0 for lexical items increases.

The final parameter of the bigram model, α_1, has the greatest effect on results. Recall that this parameter determines the probability of generating a novel bigram. Since this is the only parameter that deals with word context, it is not surprising that it has such a strong effect on segmentation accuracy. Small values of α_1 lead to solutions with fewer novel bigrams, which is achieved by oversegmenting words into smaller units. As α_1 rises, the number of proposed boundaries falls, which lowers boundary recall but increases precision. The lexicon becomes both larger and more correct. For moderate values of α_1, a good balance is achieved between oversegmentation and undersegmentation of the corpus, and both token accuracy and lexicon

[10] Our implementation is slightly different than in the original presentation of this model (Goldwater et al., 2006a), and also fixes a small bug in that implementation. Thus, the results presented here are quantitatively (though not qualitatively) different from those presented in previous papers.

[11] In the full model including utterance boundaries described in Appendix A, there is a fourth free parameter, $p_\$$. However, we found that this parameter had almost no effect on results, and kept it fixed at 0.5 for all simulations reported here.

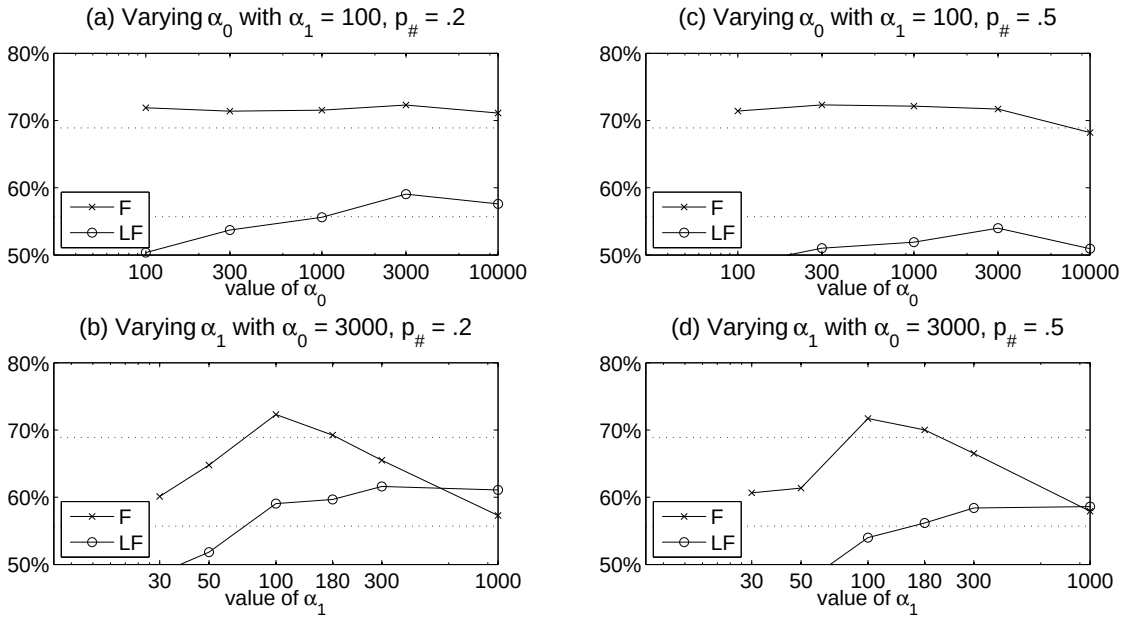

Figure 7: F_0 for words (F) and lexical items (LF) in the HDP model as a function of the different model parameters. Each plot varies either α_0 (plots (a) and (c)) or α_1 (plots (b) and (d)), while holding the other parameters fixed.

Table 6: Word segmentation accuracy of unigram and bigram systems

	Performance measure								
Model	P	R	F	BP	BR	BF	LP	LR	LF
NGS-u	67.7	**70.2**	68.9	80.6	**84.8**	82.6	52.9	51.3	52.0
MBDP-1	67.0	69.4	68.2	80.3	84.3	82.3	53.6	51.3	52.4
DP	61.9	47.6	53.8	**92.4**	62.2	74.3	57.0	**57.5**	57.2
NGS-b	68.1	68.6	68.3	81.7	82.5	82.1	54.5	57.0	55.7
HDP	**75.2**	69.6	**72.3**	90.3	80.8	**85.2**	**63.5**	55.2	**59.1**

Note: Unigram scores are from Table 1, repeated here for convenience. P, R, and F are precision, recall, and F_0 for word tokens; BP, LP, etc. are the corresponding scores for ambiguous boundaries and lexical items. Best scores are shown in bold. HDP results are with $p_\$ = .5, p_\# = .2, \alpha_0 = 3000$, and $\alpha_1 = 100$.

accuracy are high. Token accuracy in particular is dramatically higher than in the unigram model, and high-frequency words are correctly segmented far more often.

Table 6 presents the results of the HDP model using the parameter settings with highest token F_0 ($\alpha_0 = 3000, \alpha_1 = 100$, and $p_\# = 0.2$), as well as results from the only previously published model incorporating bigram dependencies, NGS-b. Results from the three unigram models in Table 1 are replicated here for comparison. Due to search, the performance of the NGS-b model is not much different from that of the NGS-u. In contrast, the HDP model performs far better than the DP model on several measures and achieves the highest segmentation accuracy of all the models tested here.[12] As illustrated in Figure 8, the segmentation found by our bigram model contains far fewer errors than the segmentation found by our unigram model, and undersegmentation is much less prevalent. Table 6 shows that our bigram model outperforms the other models on several measures, and is very close to best performance on the others. This improvement can be attributed to a large increase in boundary recall relative to the DP model, with little loss in precision. In other words, the bigram model proposes more word boundaries and is almost as accurate with those proposals.

Not only are the results of the bigram model much better than those of the basic unigram model $\mathrm{DP}(\alpha_0, P_0)$, they are qualitatively different. In the unigram model, type accuracy is higher than token accuracy, indicating many errors on frequent words. In the bigram model, the opposite is true: frequent words are much more likely to be segmented correctly, so token accuracy is higher than type accuracy. As Table 7 shows, the bigram model does make some collocation errors, but they are far less common than in the unigram model. Other kinds of errors make up a larger proportion of the errors in the bigram model. A particularly interesting kind of error is the segmentation of suffixes as individual words. The top 100 most frequent lexical items proposed by the bigram model include `z`, `s`, `IN`, `d`, and `t`, which correspond to plural, progressive, and past tense endings. Together, these items account for 2.4% of the segmented tokens. Interestingly, this is the same percentage as in the unigram model (although it is a larger proportion of incorrect tokens, due to the reduction of other errors). This suggests that incorporating word-to-word dependencies may not help to account for morphological dependencies. Incorporating a notion of morphology or syllable structure into the model (similar to the models presented by Johnson (2008)) could improve results.

[12] At the time we developed the HDP model, it outperformed all published results on the corpus used here. Since then, higher accuracy has been achieved by another Bayesian model based on similar principles (Johnson, 2008). Gambell and Yang (2006) also report much higher accuracy using a non-statistical method, but these results are based on a different corpus and input representation. See the Discussion section for further details.

you want to see the book	+ 974 what
look there's a boy with his hat	+ 847 you
and a doggie	+ 839 the
you want to lookat this	+ 815 a
lookat this	+ 455 that
have a d rink	+ 418 to/too/two
okay now	+ 399 that's
what's this	+ 389 it
what's that	+ 380 this
what isit	+ 360 okay
look canyou take itout	+ 359 and
take itout	+ 357 see
you want itin	+ 337 yeah
put that on	+ 336 what's
that	+ 336 I
yes	- 308 [z]
okay	+ 289 look
open itup	+ 270 it's
take thedoggie out	- 268 doyou
i think it will comeout	+ 268 no
let'ssee	+ 267 your/you're
yeah	+ 267 there/their
pull itout	+ 249 put
what's it	+ 247 want
look	- 229 canyou
look	+ 228 in
what's that	+ 226 wanna
get it	+ 193 one
get it	+ 191 like
get it	+ 187 here
isthat for thedoggie	+ 171 book
canyou feed itto thedoggie	- 167 sthat
feed it	+ 166 his
put itin okay	+ 165 is
okay	+ 160 do

Figure 8: Results for the HDP model with $p_{\#} = .2$, $\alpha_0 = 3000$, and $\alpha_1 = 100$: the first 35 segmented utterances (left) and the 35 most frequent lexical items (right). Fewer collocations appear than in the DP model, there are fewer errors on high-frequency words, and word frequencies match the true frequencies (Figure 4) more closely.

Table 7: Error analysis for two unigram models and the HDP bigram model

		Token errors				Lexicon errors	
Model	# Toks	Collocs	Non-wds	Placmt	# Types	Collocs	Non-wds
DP(α_0, P_0)	25677	29.3%	5.3%	3.5%	1331	30.8%	12.2%
DP(α_0, P_{true})	27503	21.4%	1.4%	1.3%	1325	13.1%	1.4%
HDP	30914	12.0%	8.0%	4.8%	1148	14.7%	21.8%

Note: Shown are the number of tokens and lexical items found by each system, and the percentage of those consisting of collocations, other items not in the true lexicon, and placement errors (words belonging to the true lexicon, but proposed in the wrong location). Parameters for the DP models were $p_\# = .5$, $\alpha_0 = 20$. The mixing constant in the DP(α_0, P_{true}) model was $\epsilon = 10^{-6}$. Parameters for the HDP model were $p_\# = .2$, $\alpha_0 = 3000$, and $\alpha_1 = 100$.

Table 8: Results of the HDP model, averaged over multiple samples

	F	LF	$-\log P(\mathbf{w})$
Samples (10 runs)	71.7 (.56)	57.1 (.85)	199370 (653)
Samples (1 run)	71.0 (.12)	56.3 (.75)	199500 (400)
MAP approx.	71.7 (.67)	58.8 (.87)	182450 (155)

Note: Token F_0 (F), lexicon F_0 (LF), and negative log posterior probability were averaged over 10 samples from independent runs of our Gibbs sampler, over 10 samples from a single run, and over 10 samples from independent runs of our MAP approximation (see text). Standard deviations are shown in parentheses.

Comparison of the bigram model to the DP(α_0, P_{true}) model is particularly enlightening. Access to the true word types gives the unigram model much higher accuracy on lexical items, but frequent items are still analyzed as collocations at a much higher rate than in the bigram model. The net result is that the bigram model scores better on token accuracy, even though it is completely unsupervised. This difference between type accuracy and token accuracy is not surprising: the contextual dependencies built into the bigram model primarily encode information about the behavior of word tokens. With even a small amount of uncertainty in the contents of the lexicon, a model that doesn't take word usage into account will have difficulty segmenting natural language. On the other hand, incorporating contextual dependencies allows the model to learn about likely sequences of words, greatly improving segmentation while also building a fairly accurate lexicon.

As in the unigram model, we performed additional simulations to examine the amount of variability in the results produced by a single sample of the bigram model and determine whether the MAP approximation would improve segmentation. Average results over ten samples are shown in Table 8. Again we find that taking samples from a single run yields less variability than taking samples from independent runs. Unlike our unigram results, the MAP approximation does seem to reduce variability, and yields significantly higher lexicon F_0 than the standard sampler ($p < .002$ according to a Wilcoxon sum-rank test). The average log posterior probability of the MAP approximation is also lower than that of the standard sampler ($p < .0005$), although segmentation accuracy is not significantly different.

6 General discussion

In this paper we have developed several computational models based on Bayesian statistics in order to explore the effects of context on word segmentation. Unlike previous work investigating how transitional probabilities or similar statistics might be used to identify boundaries, our modeling effort focuses on the problem of learning words. Changing the focus in this way brings to light the distinction between two possible assumptions about the behavior of words in natural language: the assumption that words are statistically independent units, and the assumption that words are predictive units. Our empirical and analytic results show that, for an ideal learner, adopting the first assumption will lead to undersegmentation of natural language, with many common collocations identified as single words. Assuming instead that words are predictive of each other, an ideal learner can produce far more accurate segmentations.

These results raise a number of questions about the consequences of the assumptions that were made in defining our models. In this section, we briefly discuss these assumptions, identify connections to other models, and point out ways in which our work could be extended. We close by considering the implications of our results for behavioral experiments exploring human statistical learning.

6.1 Ideal observer models of statistical learning

Our models indicate how an ideal learner provided with all the information contained in a corpus and able to evaluate all possible segmentations would choose to segment child-directed speech. There are those who would argue that human infants are in no way ideal learners – either because they are not seeking to optimize any particular objective function, or because they simply do not have the means to do so (or even come close). If that is the case, then our conclusions may be interesting from a theoretical perspective, but have little to say about human language acquisition. However, we feel that the question of whether (or in what situations) humans behave as ideal learners is still very much unresolved, and indeed is an active and growing research topic. Developing explicit ideal learner models with testable predictions, as we have done here, provides a way to investigate this question more fully in the future. In fact, we are currently engaged in research comparing the predictions of our Bayesian word segmentation models with the predictions of a number of previously proposed models in several human word segmentation experiments (Frank et al., 2007; Frank et al., in preparation)

While the issue of whether infants are ideal learners affects the extent to which the models we have presented should be taken as making predictions about infant behavior, our results are still informative as an indication of the best a learner might be expected to do with a particular corpus and set of assumptions about the structure of language. In this way, the model plays a simlar role to ideal observer analyses in visual perception, which tell us what an observer would see if they were able to optimally extract information from a stimulus (Yuille and Kersten, 2006). Indeed, the approach that we have taken here is complemented by a recent model of statistical learning for visual stimuli which was directly motivated by this kind of ideal observer analysis (Orbán et al., 2008).

Orbán et al. (2008) developed an ideal observer model to explain how people learn regularities in patterns of shapes appearing in a two-dimensional visual array. The motivation for this work was discovering how people learn the "chunks" that should be used in encoding visual scenes. The paradigm is a visual analogue of the statistical learning experiments with speech sounds that have been the focus of our analysis, with each chunk consisting of several lower-level visual features in the same way that words consist of speech sounds, and the goal of learning being the identification of these regularities through exposure. The model that Orbán et al. developed assumes that each image is generated by first activating some number of chunks, and then sampling the locations of the visual features that comprise those chunks. The total number of chunks expressed in a set of images is left free to vary, being chosen from a prior distribu-

tion. Applying probabilistic inference makes it possible to identify the chunks that were used to generate a set of images, as well as how many such chunks are necessary.

There are two basic differences between the model proposed by Orbán et al. and the model that we have presented here. The first is that the features that comprise chunks are allowed to appear in variable locations in two-dimensional space, while the sounds that comprise words need to appear in fixed order in a speech stream. The second is that an image is encoded simply through the presence or absence of these features, while a segmentation of a stream of speech sounds also carries information about the order in which the words appear, which is particularly important for our bigram model. These differences reflect the differences between the kind of stimuli that the models are designed to process, and the kinds of regularities they are intended to detect. As a consequence, the two models are not directly comparable, each being specialized to its own domain. However, the models have several similarities, including their ability to determine how many chunks or words are needed to explain a set of images or a corpus, and their foundation in the principles of Bayesian statistics. In particular, both models are based on defining a procedure for generating stimuli (either images or utterances) that is inverted by applying Bayes' rule to recover latent structure.

6.2 Online inference in word segmentation

When discussing ideal learning in relation to humans, questions about learning algorithms inevitably arise. There are two questions we are often asked with regard to the work presented here. First, are there algorithms that could be used to optimize the kinds of objective functions we propose in a more cognitively plausible way? Second, what is it about the algorithms used by Brent and Venkataraman that allows their systems to succeed despite problems with the underlying models? The algorithm we have used here assumes that the entire data set is available in memory for the learner to iterate over during learning, while Brent's and Venkataraman's algorithms operate in an online fashion, observing (and learning from) each utterance in the data set exactly once. It is worth noting that the two online algorithms are very similar, but beyond that, we have no special insight at present into why they perform as well as they do. However, in future work we may begin to address this question as we examine possible online learning algorithms for our own models.

As we mentioned earlier, there exist online algorithms for similar types of models which can be made to approximate the optimal posterior distribution with varying levels of accuracy, depending on the amount of memory they are allowed to use. The most promising kind of algorithm for online inference in our model is the particle filter (Doucet et al., 2000; Doucet et al., 2001), in which the posterior distribution is approximated by a set of samples from that distribution. These samples are then updated as new observations are made. In our model, each sample would consist of a segmentation of the corpus and samples would be updated on hearing a new utterance. The new utterance would be segmented using the lexicon associated with each sample, and those samples that produce good segmentations would be assigned higher weight. This procedure allows a good segmentation and lexicon to be inferred in an online fashion. Particle filters are becoming increasingly widespread as a means of translating Bayesian models into process models capable of making trial-by-trial predictions (Sanborn et al., 2006a; Daw and Courville, 2008; Brown and Steyvers, in press), along with other approximate methods for performing Bayesian inference (Kruschke, 2006, for example).

We plan to implement a particle filter for this model in the future and investigate how differing memory capacity, reflected in the number of samples maintained, might affect the results of segmentation under this regime. By comparing the results of our own online algorithm to those of Brent and Venkataraman, we may also gain insight into the success of those previous algorithms. In any case, we emphasize that the results presented in this paper, which assume that the learner is able to correctly identify the posterior distribution, are an important first step

in teasing apart how the outcome of learning is affected (on the one hand) by the underlying goals and assumptions of the learner, and (on the other hand) by whatever procedures are used to achieve those goals. Our analysis shows how making different assumptions about language influences the conclusions that an ideal learner should reach, and do not depend on the particular learning algorithm we used, only on the assumption that *some* algorithm is available to the learner that can closely approximate the optimal posterior distribution of the models presented.

6.3 Representational assumptions

Every computational model must make some assumptions about how the input data is to be represented. Following Brent (1999) and Venkataraman (2001), we have used an input corpus consisting of phonemically transcribed words. This choice allows us to compare our results directly to theirs, but does present some potential problems. First, it has been argued that syllables, not phonemes, are the basic sub-word level of representation from which infants begin to construct words (Swingley, 2005). While this may be the case, it is worth noting that a relatively small proportion of errors made by our models consist of proposing a boundary intra-syllabically, which indicates that assuming syllable structure as given may not be crucial. In fact, recent work suggests that learners may be able to acquire the structure of both syllables and words simultaneously, assuming a Bayesian model similar to those proposed here (Johnson, 2008). More importantly, we have shown that the kinds of errors we are most concerned about in this paper – collocation errors – cannot be solved by improving the learner's lexical model, which is essentially what a syllable-based input representation might do.

Another representational issue raised in recent computational work is how stress (or other prosodic information) might play a role in word segmentation. Gambell and Yang (2006), for example, suggest that when the input representation (in their case, syllables) is augmented with stress marks (with a single syllable in each word labeled "strong" and the others labeled "weak"), a simple rule-based segmentation strategy is sufficient. Their accuracy scores are certainly impressive (95% F_0 on word tokens), but may reflect an overly optimistic view of the information available to the learner. In particular, because the learner postulates word boundaries between any two strong syllables, and most monosyllabic words in their corpus (including function words) are labeled "strong", and English is a heavily monosyllabic language, many word boundaries are available essentially for free. It is not clear how performance would be affected under the more realistic assumption that most common monosyllabic words are in fact unstressed.[13] If stress is as easy to extract and use for segmentation as Gambell and Yang suggest, then infants' initial preference to segment via statistical cues (Thiessen and Saffran, 2003) remains a puzzle. Nevertheless, we have no doubt that stress serves as an important additional source of information for segmentation, and it is worth examining how the availability of stress cues might shed new light on the results we present here. For example, perhaps the kind of undersegmented solution found by our unigram model is sufficient to allow the learner to identify dominant stress patterns in the language, which could then be used to improve later segmentation.

While addressing such questions of cue combination is clearly important, we believe that a thorough understanding of the computational aspects of the word segmentation problem must begin with the simplest possible input before moving on to models combining multiple cues.

[13]Stress marks in Gambell and Yang's work were determined using the CMU pronouncing dictionary, with the first pronunciation used in case of ambiguity. Examining the dictionary reveals that, of the 20 most frequent words in our input corpus (*you, the, a, that, what, is, it, this, what's, to, do, look, can, that's, see, there, I, and, in, your*), all except *the* and *a* would be assigned "strong" stress marks according to this method. However, in actual usage, the remaining words except for *that*, *this*, *look*, and *see* are likely unstressed in nearly all circumstances. These words alone account for 24% of the tokens in our corpus, so changing their representation could have a large impact on results. Although Gambell and Yang used a different corpus, we imagine that the general pattern would be similar.

This progression is analogous to work in the behavioral study of statistical word segmentation, which began with stimuli containing only a single cue (the probabilistic relationships between syllables) and only later expanded to include complex stimuli containing multiple cues. The models presented here are developed within a flexible Bayesian framework that allows different components of the model (e.g., the model of lexical items or the model of context) to be independently modified. This flexibility will enable us to incorporate additional sources of information into our models in the future in order to examine some of the questions of representation and cue combination mentioned here.

A related issue of representation is the fact that we have completely abstracted away from the acoustic and phonetic variability in the input that human learners receive. While criticizing Gambell and Yang for their use of citation stress patterns, we have permitted ourselves the luxury of normalized phonemic representations. However, we emphasize two points. First, our theoretical results regarding the problem of undersegmentation in *any* unigram model do not depend on the particular choice of input representation, and suggest that in order to overcome this tendency, any additional cues extracted from the input would have to be overwhelmingly strong. Second, although our current system uses an idealized noise-free input corpus, a major advantage of statistical methods is their ability to handle noisy input in a robust way. In future work, we plan to extend the models presented here to account for variability and noise in the input, and investigate how this affects the resulting segmentation.

Finally, some readers might wonder about the effect of our choice to use a corpus in which utterance boundaries are given. While this is probably a reasonable assumption (since such boundaries can generally be determined based on pauses in the input), it is fair to ask how important the utterance boundaries are to our results. Experimental evidence suggests that human subjects' segmentation accuracy improves as utterances become shorter (i.e., as more utterance boundaries are provided) (Frank et al., 2007; Frank et al., in preparation), and very short utterances consisting of isolated words seem to be important in children's early word learning (Brent and Siskind, 2001). On the other hand, it has also been shown that word segmentation is possible even without the presence of utterance boundaries (Saffran et al., 1996a, inter alia). While we have not performed extensive tests of the effects of utterance boundaries in our models, we did run some preliminary simulations using the same corpus as in our reported results, but with all utterance boundaries removed. The results of these simulations revealed that for both the DP and HDP models, segmentation accuracy was similar to or perhaps slightly worse than the accuracy on the original corpus. (Due to randomness in the results, more extensive testing would be required to determine whether the results with and without utterance boundaries were significantly different.) The lack of much difference in results with and without utterance boundaries is somewhat surprising, given that utterance boundaries provide a certain number of word boundaries for free. However, our models do not explicitly incorporate any notion of phonotactics, which could be where much of the benefit lies in knowing utterance boundaries (i.e., because utterance boundaries allow the learner to identify phone sequences that commonly occur at word edges). It is also worth noting that the behavior of our models on continuous input is very different from that of MBDP-1 (Brent, 1999): when presented with a continuous stream of phonemes, Brent's learner will fail to find any segmentation at all. As with many of the other differences between our learners and MBDP-1, this difference is not due to differences in the probabilistic models underlying the systems, but rather to Brent's online search procedure, which requires the presence of utterance boundaries in order to begin to identify words. Other computational models, including Venkataraman's (2001), the recent phonotactic-based model of Fleck (2008), and the connectionist models of Christiansen et al. (1998) and Aslin et al. (1996), also crucially require utterance boundaries in the training corpus in order to predict word boundaries. (We note, however, that similar connectionist models could in principle perform segmentation of continuous input, given an appropriate interpretation of the output.) All of these models differ from our own in this

respect, and fail to explain how infants are able to segment words from the continuous streams of input used in experiments such as those of Saffran et al. (1996a).

6.4 Implications for behavioral research

To date, behavioral experiments of the sort exemplified by Saffran et al. (1996a) have typically used stimuli that are constructed according to a unigram model. Thus, transitions between words are random, while transitions within words are predictable and occur with higher probability. This assumption is, of course, a simplification of natural language, but has been useful for demonstrating that human learners are able to segment speech-like input on the basis of statistical regularities alone. Continuing work has examined how these kinds of statistical regularities interact with other kinds of cues such as stress and phonetic variability. However, nearly all this work is based on stimuli in which statistical regularities exist at only a single level, usually the level of syllables. (One exception is Newport et al. (in preparation), in which regularities at both the phoneme level and syllable level are considered.) Our simulations indicate that a learner who is only able to track regularities at the sub-word level faces a severe handicap when trying to segment natural language based on purely statistical information. This is because regularities in sub-word units may occur as a result of these units being grouped within words, or as a result of the words themselves being grouped within utterances. Without taking into account the larger (word-level) context, the learner must assume that all regularities are a result of groupings within words. This assumption causes undersegmentation of the input, as word-level groupings are analyzed as individual words.

The fact that regularities exist at many different levels in natural language should come as no surprise to anyone, yet our results are a reminder that it is important to consider the consequences of this hierarchical structure even for very early language acquisition tasks. Based on existing behavioral studies, we do not know whether humans are able to track and use bigram or other higher-level statistics for word segmentation; our work indicates that this would certainly be helpful, and suggests that the question is an important one to pursue. Unfortunately, designing experimental stimuli that move beyond unigrams could be too complex to be feasible within typical statistical learning paradigms. However, it might be possible to investigate this question less directly by probing adults' sensitivity to bigram frequencies, or by more thoroughly examining the nature of undersegmentation errors in young children. Peters' (1983) work on children's production errors provides a good starting point, but is necessarily limited to children who have already begun to talk, which is well beyond the age at which word segmentation begins. Our findings suggest an important role for experimental studies that would examine the possibility of widespread undersegmentation errors in the perception and representation of natural speech in preverbal infants.

7 Conclusion

In this paper, we have presented two computational models of word segmentation developed within a Bayesian ideal learner framework. The first model, which makes the assumption that words are statistically independent units, was found to severely undersegment the input corpus. Moreover, our analytical results show that this kind of undersegmentation is unavoidable for any ideal learner making the same assumption of independence between words. In contrast, our second model demonstrates that a more sophisticated statistical learner that takes into account dependencies both at the sub-word and word level is able to produce much more accurate (adult-like) segmentations. These results do not yet provide direct evidence of whether infants are able to take context into account during early segmentation, but do provide specific predictions that can be tested in future research. In particular, we envision three competing hypotheses that these models may help to tease apart. First, infant learners may bear no re-

semblance to ideal learners in this task. If this is the case, then we would expect the predictions of our models to differ from human performance even on simple stimuli where words are independent. This hypothesis can therefore be tested using standard statistical learning paradigms, as we are currently engaged in doing. Second, infants may approximate ideal learners, but be unable to process statistical information at multiple levels early on. We would then expect to find widespread undersegmentation in early speech perception, a phenomenon which has yet to be examined. If this hypothesis is correct, we would also need to complete a developmental story explaining how additional cues (including perhaps word-level dependencies) are later incorporated to correct the initial undersegmentation. Finally, infants may be best modeled as ideal learners who are able to process sophisticated statistical information, including contextual cues, even very early in learning. In this case, early word representations would be more adult-like, and might require less modification after stress and other cues become more important later in infancy. Our work helps to clarify the differences between these three positions, and we hope that it will provide a source of inspiration for future experimental and computational studies examining the evidence for each.

References

E. Aarts and J. Korst. 1989. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley, New York.

D. Aldous. 1985. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII—1983*, pages 1–198. Springer, Berlin.

J. Allen and M. Christiansen. 1996. Integrating multiple cues in word segmentation: a connectionist model using hints. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society*, pages 370–375, Mahwah, NJ. Lawrence Erlbaum.

J. R. Anderson. 1991. The adaptive nature of human categorization. *Psychological Review*, 98(3):409–429.

R. Ando and L. Lee. 2000. Mostly-unsupervised statistical segmentation of Japanese: Application to kanji. In *Proceedings of ANLP-NAACL*.

R. Arratia, A. D. Barbour, and S. Tavare. 1992. Poisson process approximations for the Ewens sampling formula. *The Annals of Applied Probability*, 2:519–535.

R. Aslin, J. Woodward, N. LaMendola, and T. Bever. 1996. Models of word segmentation in fluent maternal speech to infants. In J. Morgan and K. Demuth, editors, *Signal to Syntax*, pages 117–134. Lawrence Erlbaum Associates, Mahwah, NJ.

R. Aslin, J. Saffran, and E. Newport. 1998. Computation of conditional probability statistics by 8-month-old infants. *Psychological Science*, 9:321–324.

E. Batchelder. 2002. Bootstrapping the lexicon: A computational model of infant speech segmentation. *Cognition*, 83:167–206.

J. Bernardo and M. Smith. 1994. *Bayesian Theory*. Wiley, New York.

N. Bernstein-Ratner. 1987. The phonology of parent-child speech. In K. Nelson and A. van Kleeck, editors, *Children's Language*, volume 6. Erlbaum, Hillsdale, NJ.

C. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer, New York.

D. Blackwell and J. MacQueen. 1973. Ferguson distributions via Pólya urn schemes. *Annals of Statistics*, 1:353–355.

D. Blei, A. Ng, and M. Jordan. 2002. Latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 14*.

D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. 2004. Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems 16*.

M. Brent and T. Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61:93–125.

M. Brent and J. Siskind. 2001. The role of exposure to isolated words in early vocabulary development. *Cognition*, 81(2):B33–44.

M. Brent. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34:71–105.

S.D. Brown and M. Steyvers. in press. Detecting and predicting changes. *Cognitive Psychology*.

P. Cairns and R. Shillcock. 1997. Bootstrapping word boundaries: A bottom-up corpus-based approach to speech segmentation. *Cognitive Psychology*, 33:111–153.

S. F. Chen and J. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Center for Research in Computing Technology, Harvard University.

M. Christiansen, J. Allen, and M. Seidenberg. 1998. Learning to segment speech using multiple cues: A connectionist model. *Language and Cognitive Processes*, 13:221–268.

P. Cohen and N. Adams. 2001. An algorithm for segmenting categorical timeseries into meaningful episodes. In *Proceedings of the Fourth Symposium on Intelligent Data Analysis*.

S. Creel, E. Newport, and R. Aslin. 2004. Distant melodies: Statistical learning of non-adjacent dependencies in tone sequences. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(5):1119–1130.

M. Creutz and K. Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the Workshop on Morphological and Phonological Learning at ACL '02*.

Nathaniel Daw and Aaron Courville. 2008. The pigeon as particle filter. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 369–376. MIT Press, Cambridge, MA.

C. de Marcken. 1995. The unsupervised acquisition of a lexicon from continuous speech. Technical report, Massachusetts Institute of Technology. A.I. Memo No. 1558.

A. Doucet, C. Andrieu, and S. Godsill. 2000. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.

A. Doucet, N. de Freitas, and N. Gordon. 2001. *Sequential Monte Carlo Methods in Practice.* Springer, New York.

M. Dowman. 2000. Addressing the learnability of verb subcategorizations with Bayesian inference. In L. Gleitman and A. Joshi, editors, *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society*, Mahwah, NJ. Lawrence Erlbaum Associates.

C. Elkan. 2006. Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution. In *Proceedings of the 23rd International Conference on Machine Learning*.

T. M. Ellison. 1994. The iterative learning of phonological constraints. *Computational Linguistics*, 20(3).

J. Elman. 1990. Finding structure in time. *Cognitive Science*, 14:179–211.

M. Escobar and M. West. 1995. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.

H. Feng, K. Chen, X. Deng, and W. Zheng. 2004. Accessor variety criteria for Chinese word extraction. *Computational Lingustics*, 30(1).

S. Ferguson. 1973. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1:209–230.

J. Finkel, T. Grenager, and C. Manning. 2007. The infinite tree. In *Proceedings of the ACL*.

J. Fiser and R. Aslin. 2002. Statistical learning of new visual feature combinations by infants. *Proceedings of the National Academy of Sciences*, 99:15822–15826.

M. Fleck. 2008. Lexicalized phonotactic word segmentation. In *Proceedings of ACL-08: HLT*, pages 130–138, Columbus, Ohio. Association for Computational Linguistics.

M. Frank, S. Goldwater, V. Mansinghka, T. Griffiths, and J. Tenenbaum. 2007. Modeling human performance on statistical word segmentation tasks. In *Proceedings of CogSci*.

M. C. Frank, S. Goldwater, T. Griffiths, E. Newport, R. Aslin, and J. Tenenbaum. in preparation. Modeling human performance in statistical word segmentation.

T. Gambell and C. Yang. 2006. Word segmentation: Quick but not dirty. Unpublished manuscript, available at http://www.ling.upenn.edu/ ycharles/papers/quick.pdf.

A. Gelman, J. Carlin, H. Stern, and D. Rubin. 2004. *Bayesian Data Analysis*. Chapman & Hall/CRC, New York.

S. Geman and D. Geman. 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

W.R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. 1996. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, Suffolk.

J. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27:153–198.

S. Goldwater and M. Johnson. 2004. Priors in Bayesian learning of phonological rules. In *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON '04)*.

S. Goldwater, T. Griffiths, and M. Johnson. 2006a. Contextual dependencies in unsupervised word segmentation. In *Proceedings of COLING/ACL*, Sydney.

S. Goldwater, T. Griffiths, and M. Johnson. 2006b. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*.

S. Goldwater, T. Griffiths, and M. Johnson. 2007. Distributional cues to word segmentation: Context is important. In *Proceedings of the 31st Boston University Conference on Language Development*.

S. Goldwater. 2006. *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis, Brown University.

R. Gómez and J. Maye. 2005. The developmental trajectory of nonadjacent dependency learning. *Infancy*, 7:183–206.

Z. Harris. 1954. Distributional structure. *Word*, 10:146–162.

Z. Harris. 1955. From phoneme to morpheme. *Language*, 31:190–222.

W. Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.

E. Johnson and P. Jusczyk. 2001. Word segmentation by 8-month-olds: When speech cues count more than statistics. *Journal of Memory and Language*, 44:548–567.

M. Johnson, T. Griffiths, and S. Goldwater. 2007. Adaptor grammars: a framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems 19*.

M. Johnson. 2008. Using adapter grammars to identify synergies in the unsupervised learning of linguistic structure. In *Proceedings of ACL*.

P. Jusczyk, E. Hohne, and A. Bauman. 1999a. Infants' sensitivity to allophonic cues for word segmentation. *Perception and Psychophysics*, 61(8):1465–1476.

P. Jusczyk, D. Houston, and M. Newsome. 1999b. The beginnings of word segmentation in English-learning infants. *Cognitive Psychology*, 39:159–207.

P. Jusczyk. 1999. How infants begin to extract words from speech. *Trends in Cognitive Sciences*, 3(9).

R. Kneser and H. Ney. 1995. Improved backing-off for n-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

J. Kruschke. 2006. Locally Bayesian learning with applications to retrospective revaluation and highlighting. *Psychological Review*, 113(4):677–699.

P. Liang, S. Petrov, M. Jordan, and D. Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of (EMNLP/CoNLL)*.

A. Lo. 1984. On a class of Bayesian nonparametric estimates. *Annals of Statistics*, 12:351–357.

D. MacKay and L. Bauman Peto. 1994. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(1).

B. MacWhinney and C. Snow. 1985. The child language data exchange system. *Journal of Child Language*, 12:271–296.

R. Madsen, D. Kauchak, and C. Elkan. 2005. Modeling word burstiness using the Dirichlet distribution. In *Proceedings of the 22nd International Conference on Machine Learning*.

D. Marr. 1982. *Vision: A Computational Approach*. Freeman & Co., San Francisco.

S. Mattys, P. Jusczyk, P. Luce, and J. Morgan. 1999. Phonotactic and prosodic effects on word segmentation in infants. *Cognitive Psychology*, 38:465–494.

N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. 1953. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.

T. Mintz. 2002. Category induction from distributional cues in an artificial language. *Memory and Cognition*, 30:678–686.

J. Morgan, K. Bonamo, and L. Travis. 1995. Negative evidence on negative evidence. *Developmental Psychology*, 31:180–197.

D. Navarro, T. Griffiths, M. Steyvers, and M. Lee. 2006. Modeling individual differences using Dirichlet processes. *Journal of Mathematical Psychology*.

R. Neal. 1993. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, Department of Computer Science.

R. Neal. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265.

E. Newport and R. Aslin. 2004. Learning at a distance I. Statistical learning of non-adjacent dependencies. *Cognitive Psychology*, 48:127–162.

E. Newport, D. Weiss, R. Aslin, and L. Wonnacott. in preparation. Statistical learning in speech by infants: Syllables or segments?

G. Orbán, J. Fiser, R. Aslin, and M. Lengyel. 2008. Bayesian learning of visual chunks by human observers. *Proceedings of the National Academy of Sciences*, 105:2745– 2750.

A. Peters. 1983. *The Units of Language Acquisition*. Cambridge University Press, New York.

C. Rasmussen. 2000. The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems 12*.

J. Rissanen. 1989. *Stochastic Complexity and Statistical Inquiry*. World Scientific Co., Singapore.

J. Saffran, R. Aslin, and E. Newport. 1996a. Statistical learning in 8-month-old infants. *Science*, 274:1926–1928.

J. Saffran, E. Newport, and R. Aslin. 1996b. Word segmentation: the role of distributional cues. *Journal of Memory and Language*, 35:606–621.

A. Sanborn, T. Griffiths, and D. Navarro. 2006a. A more rational model of categorization. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*.

A. N. Sanborn, T. L. Griffiths, and D. J. Navarro. 2006b. A more rational model of categorization. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, Mahwah, NJ. Erlbaum.

L. Schulz, E. Bonawitz, and T. Griffiths. 2007. Can being scared make your tummy ache? naive theories, ambiguous evidence and preschoolers' causal inferences. *Developmental Psychology*, 43:1124–1139.

D. Swingley. 2005. Statistical clustering and the contents of the infant vocabulary. *Cognitive Psychology*, 50:86–132.

Y. Teh, M. Jordan, M. Beal, and D. Blei. 2005. Hierarchical Dirichlet processes. In *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA.

Y. Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, National University of Singapore, School of Computing.

E. Thiessen and J. Saffran. 2003. When cues collide: Use of stress and statistical cues to word boundaries by 7- to 9-month-old infants. *Developmental Psychology*, 39(4):706–716.

E. Thiessen and J. Saffran. 2004. Spectral tilt as a cue to word segmentation in infancy and adulthood. *Perception and Psychophysics*, 66(5):779–791.

J. Toro, S. Sinnett, and S. Soto-Faraco. 2005. Speech segmentation by statistical learning depends on attention. *Cognition*, 97:B25 –B34.

A. Venkataraman. 2001. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):351–372.

F. Wood, S. Goldwater, and M. Black. 2006. A non-parametric Bayesian approach to spike sorting. In *IEEE Engineering Medicine Biologicial Systems*.

F. Xu and J. Tenenbaum. 2007. Word learning as Bayesian inference. *Psychological Review*, 114:245–272.

A. Yuille and D. Kersten. 2006. Vision as Bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, 10:301–308.

G. Zipf. 1932. *Selective Studies and the Principle of Relative Frequency in Language*. Harvard University Press, Cambridge, MA.

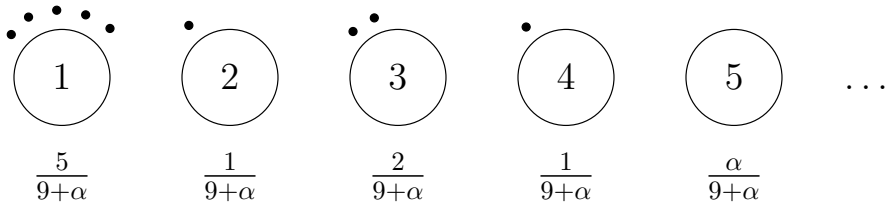

Figure 9: The Chinese restaurant process. Black dots indicate the seating arrangement of the first nine customers. Below each table is $P(z_{10} = k \,|\, \mathbf{z}_{-10})$.

A Model definitions

A.1 Unigram model

Recall that our unigram model generates the ith word in the corpus, w_i, according to the following distribution:

(1) $P(w_i \text{ is novel}) = \frac{\alpha_0}{n+\alpha_0}$, $P(w_i \text{ is not novel}) = \frac{n}{n+\alpha_0}$

(2) a. $P(w_i = x_1 \ldots x_M \,|\, w_i \text{ is novel}) = p_\#(1-p_\#)^{M-1} \prod_{j=1}^{M} P(x_j)$

b. $P(w_i = \ell \,|\, w_i \text{ is not novel}) = \frac{n_\ell}{n}$

In this section, we first show how this model can be viewed in terms of two models that are well-known in the nonparametric Bayesian statistical literature: the Chinese restaurant process (Aldous, 1985) and the Dirichlet process (Ferguson, 1973). We then provide full details of the model we used to account for utterance boundaries, and describe our Gibbs sampling algorithm.

A.1.1 The Chinese restaurant process

The Chinese restaurant process (CRP) is a stochastic process that creates a partition of items into groups. Imagine a restaurant containing an infinite number of tables, each with infinite seating capacity. Customers enter the restaurant and seat themselves. Each customer sits at an occupied table with probability proportional to the number of people already seated there, and at an unoccupied table with probability proportional to some constant α. That is, if z_i is the number of the table chosen by the ith customer and $\mathbf{z}_{-i}$ are the tables chosen by the customers preceding the ith customer, then

$$P(z_i = k \,|\, \mathbf{z}_{-i}) = \begin{cases} \frac{n_k^{(\mathbf{z}_{-i})}}{i-1+\alpha} & 1 \le k \le K(\mathbf{z}_{-i}) \\ \frac{\alpha}{i-1+\alpha} & k = K(\mathbf{z}_{-i}) + 1 \end{cases} \tag{3}$$

where $n_k^{(\mathbf{z}_{-i})}$ is the number of customers already sitting at table k, $K(\mathbf{z}_{-i})$ is the total number of occupied tables in $\mathbf{z}_{-i}$, and $\alpha \ge 0$ is a parameter of the process determining how "spread out" the customers become. Higher values of α mean that more new tables will be occupied relative to the number of customers, leading to a more uniform distribution of customers across tables. The first customer by definition sits at the first table, so this distribution is well-defined even when $\alpha = 0$. See Figure 9 for an illustration.

Under the Chinese restaurant process model, the probability of a particular sequence of table assignments for n customers is (for $\alpha > 0$)

$$P(\mathbf{z}) \quad = \quad \frac{\Gamma(\alpha)}{\Gamma(n+\alpha)} \cdot \alpha^{K(\mathbf{z})} \cdot \prod_{k=1}^{K(\mathbf{z})} (n_k^{(\mathbf{z})} - 1)! \tag{4}$$

The Gamma function appearing in Equation 4 is defined as $\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}du$ for $x > 0$, and is a generalized factorial function: $\Gamma(x) = (x-1)!$ for positive integer x, and $\Gamma(x) = (x-1)\Gamma(x-1)$ for any $x > 0$.

Note that this distribution is the same as the distribution defined by Steps 1 and 2b of our unigram model. It assigns each outcome to a group, but does not distinguish the groups in any interesting way. However, we can extend the Chinese restaurant metaphor by imagining that the first customer to sit at each table opens a fortune cookie containing a single word, and this word then provides a label for the table. The words in the cookies are generated by the distribution P_0, and the number of customers at each table corresponds to the number of occurrences of that word in the corpus. Thus, this model can be viewed as a two-stage restaurant (Goldwater et al., 2006b; Goldwater, 2006), where P_0 generates word types and the CRP generates frequencies for those types.

In the two-stage restaurant, the probability of the ith word in a sequence, given the previous labels and table assignments, can be found by summing over all the tables labeled with that word:

$$\begin{aligned}
P(w_i = \ell \,|\, \mathbf{z}_{-i}, \boldsymbol{\ell}(\mathbf{z}_{-i}), \alpha) &= \sum_{k=1}^{K(\mathbf{z}_{-i})+1} P(w_i = \ell \,|\, z_i = k, \boldsymbol{\ell}(\mathbf{z}_{-i}))P(z_i = k \,|\, \mathbf{z}_{-i}, \alpha) \\
&= \sum_{k=1}^{K(\mathbf{z}_{-i})} P(w_i = \ell \,|\, z_i = k, \ell_k)P(z_i = k \,|\, \mathbf{z}_{-i}, \alpha) \\
&\quad + P(w_i = \ell \,|\, z_i = K(\mathbf{z}_{-i}) + 1)P(z_i = K(\mathbf{z}_{-i}) + 1) \,|\, \mathbf{z}_{-i}, \alpha) \\
&= \sum_{k=1}^{K(\mathbf{z}_{-i})} I(\ell_k = \ell)\frac{n_k^{(\mathbf{z}_{-i})}}{i-1+\alpha} + P_0(\ell)\frac{\alpha}{i-1+\alpha} \\
&= \frac{n_\ell^{(\mathbf{w}_{-i})} + \alpha P_0(\ell)}{i-1+\alpha} \qquad (5)
\end{aligned}$$

where $\boldsymbol{\ell}(\mathbf{z}_{-i})$ are the labels of all the tables in $\mathbf{z}_{-i}$ (with ℓ_k being the label of table k), $I(.)$ is an indicator function taking on the value 1 when its argument is true and 0 otherwise, and $n_\ell^{(\mathbf{w}_{-i})}$ is the number of previous occurrences of ℓ in $\mathbf{w}_{-i}$ (i.e. the number of assignments in $\mathbf{z}_{-i}$ to tables labeled with ℓ). The probability of w_i conditioned only on the previously observed words (and the hyperparameter α) is also given by Equation 5, because

$$\begin{aligned}
P(w_i = \ell \,|\, \mathbf{w}_{-i}, \alpha) &= \sum_{\{\mathbf{z}_{-i}, \boldsymbol{\ell}(\mathbf{z}_{-i})\}} P(w_i = \ell \,|\, \mathbf{z}_{-i}, \boldsymbol{\ell}(\mathbf{z}_{-i}), \alpha)P(\mathbf{z}_{-i}, \boldsymbol{\ell}(\mathbf{z}_{-i}) \,|\, \alpha) \\
&= \sum_{\{\mathbf{z}_{-i}, \boldsymbol{\ell}(\mathbf{z}_{-i})\}} \frac{n_\ell^{(\mathbf{w}_{-i})} + \alpha P_0(\ell)}{i-1+\alpha} P(\mathbf{z}_{-i}, \boldsymbol{\ell}(\mathbf{z}_{-i}) \,|\, \alpha) \\
&= \frac{n_\ell^{(\mathbf{w}_{-i})} + \alpha P_0(\ell)}{i-1+\alpha} \sum_{\{\mathbf{z}_{-i}, \boldsymbol{\ell}(\mathbf{z}_{-i})\}} P(\mathbf{z}_{-i}, \boldsymbol{\ell}(\mathbf{z}_{-i}) \,|\, \alpha) \\
&= \frac{n_\ell^{(\mathbf{w}_{-i})} + \alpha P_0(\ell)}{i-1+\alpha} \qquad (6)
\end{aligned}$$

Computing the distribution over words in this way makes it clear that the probability of observing a particular word is a sum over the probability of generating that word as an old word or as a new word. That is, even words that have been observed before may be generated again by P_0 (although, in general, this probability will be low compared to the probability of generating a repeated word by sitting at a previously occupied table).

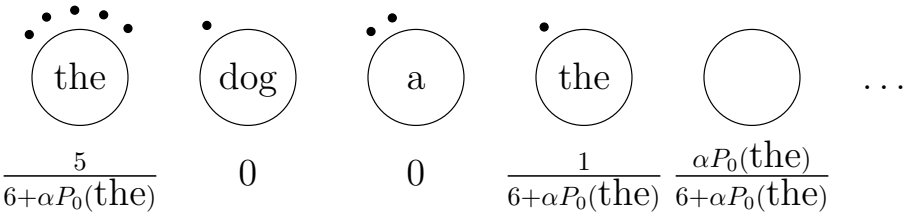

Figure 10: The two-stage restaurant. Each label ℓ_k is shown on table k. Black dots indicate the number of occurrences of each label in $\mathbf{w}_{-10}$. Below each table is $P(z_{10} = k \,|\, w_{10} = \text{'the'}, \mathbf{z}_{-10}, \boldsymbol{\ell}(\mathbf{z}_{-10}), \omega)$. Under this seating arrangement, $P(w_{10} = \text{'the'}) = \frac{6+\alpha P_0(\text{the})}{9+\alpha}$, $P(w_{10} = \text{'dog'}) = \frac{1+\alpha P_0(\text{dog})}{9+\alpha}$, $P(w_{10} = \text{'a'}) = \frac{2+\alpha P_0(\text{a})}{9+\alpha}$, and for any other word ℓ, $P(w_{10} = \ell) = \frac{\alpha P_0(\ell)}{9+\alpha}$.

In the models described in this paper, we define P_0 as a distribution over an infinite number of outcomes (all possible strings over a fixed alphabet). However, we note that it is also possible to define P_0 as a finite distribution. In the case where P_0 is a K-dimensional multinomial distribution with parameters $\boldsymbol{\omega}$, Equation 5 reduces to a K-dimensional Dirichlet($\alpha\boldsymbol{\omega}$)-multinomial model. When P_0 is a distribution over an infinite set, as in this paper, the number of different word types that will be observed in a corpus is not fixed in advance. Rather, new word types can be generated "on the fly" from an infinite supply. In general, the number of different word types observed in a corpus will slowly grow as the size of the corpus grows.

A.1.2 The Dirichlet process

Models whose complexity grows with the size of the data are referred to in the statistical literature as *infinite* or *nonparametric*, having the property that, as the data size grows to infinity, they are able to model any arbitrary probability distribution. In this section we show that the two-stage CRP model is equivalent to a standard nonparametric statistical model known as the Dirichlet process (DP).

Rather than providing a rigorous definition of the Dirichlet process here, we only attempt to give some intuition. The interested reader may refer to one of several recent papers for further exposition (Neal, 2000; Navarro et al., 2006; Teh et al., 2005). The DP is a distribution over distributions: each sample G from a DP is a distribution over a countably infinite set of outcomes. The set of outcomes over which G is defined (the *support* of G), and the relative probabilities of those outcomes, are determined by the two parameters of the DP, G_0 and α. G_0 (the *base distribution*) is a probability distribution whose support (of up to uncountably infinite size) is a superset of the support of G. For example, G_0 could be a normal distribution (over the uncountably infinite set of real numbers) or P_0 (the distribution over the countably infinite set of possible strings Σ^+ defined as part of our unigram model). The probability of any particular outcome under G_0 is the probability of that outcome being in the support of G. So, if G_0 is normal, most of the possible outcomes of G will be numbers near the mean of G_0. If $G_0 = P_0$, most of the possible outcomes of G will be relatively short strings. Given a set of outcomes for G determined by G_0, the *concentration parameter* α of the DP determines the variance of G: how uniform (or skewed) is the distribution over its possible outcomes.

It is straightforward to define a unigram language model using the DP:

$$\begin{aligned} w_i \,|\, G &\sim G \\ G \,|\, \alpha, P_0 &\sim \text{DP}(\alpha, P_0) \end{aligned} \tag{7}$$

where the $\sim$ should be read as "is distributed according to". This formulation makes the distribution G sampled from the DP explicit. Implementing such a model is not possible since

G is an infinite distribution. However, what we really want is the conditional distribution $P(w_i \,|\, \mathbf{w}_{-i}, \alpha, P_0)$, which can be found by integrating over all possible values of G:

$$P(w_i \,|\, \mathbf{w}_{-i}, \alpha, P_0) = \int P(w_i \,|\, G) P(G \,|\, \mathbf{w}_{-i}, \alpha, P_0)\ dG$$

This integration results in the following conditional distribution (Blackwell and MacQueen, 1973):

$$w_i \,|\, \mathbf{w}_{-i}, \alpha, P_0 \ \sim\ \frac{1}{i-1+\alpha} \sum_{j=1}^{i-1} \delta(w_j) + \frac{\alpha}{i-1+\alpha} P_0 \tag{8}$$

where $\delta(w_j)$ is a distribution with all its mass at w_j. Rewriting this distribution as a probability mass function makes clear the equivalence between the DP language model and the two-stage CRP language model in (5):

$$\begin{aligned} P(w_i = \ell \,|\, \mathbf{w}_{-i}, \alpha, P_0) &= \frac{1}{i-1+\alpha} \sum_{j=1}^{i-1} I(w_j = \ell) + \frac{\alpha}{i-1+\alpha} P_0(\ell) \\ &= \frac{n_\ell^{(\mathbf{w}_{-i})} + \alpha P_0(\ell)}{i-1+\alpha} \end{aligned} \tag{9}$$

The use of the Dirichlet process for learning linguistic structure, as described here, is a very recent development. More typically, the DP has been used as a prior in infinite mixture models (Lo, 1984; Escobar and West, 1995; Neal, 2000), where each table represents a mixture component, and the data points at each table are assumed to be generated from some parameterized distribution. Technically, our DP language model can be viewed as a mixture model where each table is parameterized by its label ℓ_k, and $P(w_i \,|\, \ell_{z_i}) = I(w_i = \ell_{z_i})$, so every data point in a single mixture component is identical. Previous applications of DP mixture models use more complex distributions to permit variation in the data within components. Usually Gaussians are used for continuous data (Rasmussen, 2000; Wood et al., 2006) and multinomials for discrete data (Blei et al., 2002; Navarro et al., 2006). In the area of language modeling, a number of researchers have described models similar to our DP model but with a fixed finite model size (MacKay and Peto, 1994; Elkan, 2006; Madsen et al., 2005). Some of the earliest work using DPs and their extensions for language-related tasks focused on modeling semantic content rather than linguistic structure (Blei et al., 2004). Our own earlier publications describing the models in this paper (Goldwater et al., 2006a; Goldwater, 2006; Goldwater et al., 2007) are, to our knowledge, the first to describe the application of DPs to the problem of structure induction. More recently, a number of papers have been published showing how to use DPs and hierarchical DPs (see below) for learning syntax (Johnson et al., 2007; Liang et al., 2007; Finkel et al., 2007).

A.1.3 Modeling utterance boundaries

Our model as described so far accounts for the number of times each word appears in the data. Since the input corpus used in our experiments also includes utterance boundaries, these must be accounted for in the model as well. This is done by assuming that each utterance is generated as follows:

1. Decide whether the next word will end the utterance or not.
2. Choose the identity of that word according to the DP model.
3. If the current word is not utterance-final, return to Step 1.

The proportion of words that are utterance-final is determined by a binomial distribution with parameter $p_\$$. Of course, we do not know in advance what the value of $p_\$$ should be, so we assume that this parameter is drawn from a symmetric Beta($\frac{\rho}{2}$) prior (Gelman et al., 2004). The predictive distribution of u_i (the binary variable determining whether the ith word is utterance-final or not) can be found by integrating over $p_\$$:

$$\begin{aligned} P(u_i = 1 \,|\, \mathbf{u}_{-i}, \rho) &= \int P(u_i = 1 \,|\, p_\$) P(p_\$ \,|\, \mathbf{u}_{-i}, \rho)\; dp_\$ \\ &= \frac{n_\$ + \frac{\rho}{2}}{i - 1 + \rho} \end{aligned} \tag{10}$$

where $n_\$$ is the number of utterance-final words (i.e., the number of utterances) in the first $i-1$ words. We do not show the derivation of this integral here; it is a standard result in Bayesian statistics (Gelman et al., 2004; Bernardo and Smith, 1994).

A.1.4 Inference

Our inference procedure is a Gibbs sampler that repeatedly resamples the value of each possible word boundary location in the corpus, conditioned on the current values of all other boundary locations. In a single iteration, each possible boundary is considered exactly once. It is possible to show that as the number of iterations through the training data increases, the distribution of samples (i.e., segmentations) approaches the model's posterior distribution, no matter what the initial sample was. Thus, when we run the Gibbs sampler, we discard the first several thousand iterations through the training data (this is called the "burn in" period) to allow the sampler time to begin producing samples that are approximately distributed according to the posterior distribution. Details of the burn-in procedure and other practical issues are described below; we first provide a formal description of the sampler.

Our Gibbs sampler considers a single possible word boundary location b_j at a time, so each sample is from a set of two hypotheses (sequences of words with utterance boundaries), h_1 and h_2, as illustrated in Figure 11. These hypotheses contain all the same word boundaries except at the one position under consideration, where h_2 has a boundary and h_1 does not. We can write $h_1 = \beta w_1 \gamma$ and $h_2 = \beta w_2 w_3 \gamma$, where $w_1 = w_2.w_3$, and β and γ are the sequences of words to the left and right of the area under consideration. We sample the value of b_j using the following equations:

$$P(b_j = 0 \,|\, h^-, d) = \frac{P(h_1 \,|\, h^-, d)}{P(h_1 \,|\, h^-, d) + P(h_2 \,|\, h^-, d)} \propto P(h_1 \,|\, h^-, d) \tag{11}$$

$$P(b_j = 1 \,|\, h^-, d) = \frac{P(h_2 \,|\, h^-, d)}{P(h_1 \,|\, h^-, d) + P(h_2 \,|\, h^-, d)} \propto P(h_2 \,|\, h^-, d) \tag{12}$$

where d is the observed (unsegmented) data and h^- consists of all of the words shared by the two hypotheses (i.e., the words in β and γ). Since we only care about the *relative* probabilities of the two outcomes, we can ignore the denominators and compute only the quantities in the numerators. Note that

$$\begin{aligned} P(h_1 \,|\, h^-, d) &= \frac{P(d \,|\, h_1, h^-) P(h_1 \,|\, h^-)}{P(d \,|\, h^-)} \\ &= \frac{P(h_1 \,|\, h^-)}{P(d \,|\, h^-)} \end{aligned} \tag{13}$$

for any h_1 that is consistent with the observed data, and similarly for h_2. Substituting into Equations 11 and 12 yields

$$P(b_j = 0 \,|\, h^-, d) \propto P(h_1 \,|\, h^-) \tag{14}$$

$$P(b_j = 1 \,|\, h^-, d) \propto P(h_2 \,|\, h^-) \tag{15}$$

	Current segmentation	Hypothesis 1	Hypothesis 2
(1)	↓		
	yuw.anttu.si.D6bUk lUk.D*z6b7.w.IThIz.h.&t &nd.6dOgi yu.wanttu.lUk.&t.DIs	**yuw**.anttu.si.D6bUk lUk.D*z6b7.w.IThIz.h.&t &nd.6dOgi yu.wanttu.lUk.&t.DIs	**y.uw**.anttu.si.D6bUk lUk.D*z6b7.w.IThIz.h.&t &nd.6dOgi yu.wanttu.lUk.&t.DIs
		$P(h_1\|h^-) = \frac{\alpha_0 P_0(\texttt{yuw})}{18+\alpha_0}$ (∗)	$P(h_2\|h^-) = \frac{\alpha_0 P_0(\texttt{y})}{18+\alpha_0} \cdot \frac{\alpha_0 P_0(\texttt{uw})}{19+\alpha_0}$
(2)	↓		
	yuw.anttu.si.D6bUk lUk.D*z6b7.w.IThIz.h.&t &nd.6dOgi yu.wanttu.lUk.&t.DIs	**yuw**.anttu.si.D6bUk lUk.D*z6b7.w.IThIz.h.&t &nd.6dOgi yu.wanttu.lUk.&t.DIs	**yu.w**.anttu.si.D6bUk lUk.D*z6b7.w.IThIz.h.&t &nd.6dOgi yu.wanttu.lUk.&t.DIs
		$P(h_1\|h^-) = \frac{\alpha_0 P_0(\texttt{yuw})}{18+\alpha_0}$	$P(h_2\|h^-) = \frac{1+\alpha_0 P_0(\texttt{yu})}{18+\alpha_0} \cdot \frac{1+\alpha_0 P_0(\texttt{w})}{19+\alpha_0}$ (∗)
(3)	↓		
	yu.w.anttu.si.D6bUk lUk.D*z6b7.w.IThIz.h.&t &nd.6dOgi yu.wanttu.lUk.&t.DIs	yu.**wanttu**.si.D6bUk lUk.D*z6b7.w.IThIz.h.&t &nd.6dOgi yu.wanttu.lUk.&t.DIs	yu.**w.anttu**.si.D6bUk lUk.D*z6b7.w.IThIz.h.&t &nd.6dOgi yu.wanttu.lUk.&t.DIs
		$P(h_1\|h^-) = \frac{1+\alpha_0 P_0(\texttt{wanttu})}{18+\alpha_0}$ (∗)	$P(h_2\|h^-) = \frac{1+\alpha_0 P_0(\texttt{w})}{18+\alpha_0} \cdot \frac{\alpha_0 P_0(\texttt{anttu})}{19+\alpha_0}$

Figure 11: An example illustrating our Gibbs sampling algorithm. In each of the three steps shown, a single potential boundary location (indicated with ↓) is considered. The probabilities of the words that differ (shown in bold) are computed conditioned on the remaining words in each hypothesis according to Equation 9. Then, one of the two hypotheses is chosen at random, with probability proportional to the ratio of these probabilities. We indicate the hypothesis chosen at each step in this example by (∗). Note that in the probability computation for the hypotheses in which a boundary is proposed, the denominator for the factor corresponding to the second word is incremented by one to take into account the fact that the first word is now part of the conditioning environment.

where the conditioning on d has disappeared. Calculating these quantities is now straightforward. We have

$$\begin{aligned} P(h_1 \,|\, h^-) &= P(w_1 \,|\, h^-)\, P(u_{w_1} \,|\, h^-) \\ &= \frac{n_{w_1}^{(h^-)} + \alpha_0 P_0(w_1)}{n^- + \alpha_0} \cdot \frac{n_u^{(h^-)} + \frac{\rho}{2}}{n^- + \rho} \end{aligned} \tag{16}$$

where n^- is the number of words in h^- and $n_u^{(h^-)} = n_\$^{(h^-)}$ if w_1 is utterance-final and $n^- - n_\$^{(h^-)}$ otherwise. The first factor in the second line follows from Equation 9 and the fact that the DP model is *exchangeable*: the probability of a particular sequence of words does not depend on the order of the words in that sequence (Aldous, 1985). In other words, all permutations of the sequence have equal probability. We can therefore compute the probability of any word in the sequence conditioned on all the other words by treating the word in question as if it were the last word to be generated, and applying Equation 9. The second factor similarly follows from Equation 10, since the Beta-binomial model is also exchangeable.

The posterior probability of h_2 can be computed in a similar fashion, as

$$\begin{aligned} &P(h_2 \,|\, h^-) \\ &\quad= P(w_2, w_3 \,|\, h^-) \\ &\quad= P(w_2 \,|\, h^-)\, P(u_{w_2} \,|\, h^-)\, P(w_3 \,|\, w_2, h^-)\, P(u_{w_3} \,|\, u_{w_2}, h^-) \\ &\quad= \frac{n_{w_2}^{(h^-)} + \alpha_0 P_0(w_2)}{n^- + \alpha_0} \cdot \frac{n^- - n_\$^{(h^-)} + \frac{\rho}{2}}{n^- + \rho} \\ &\qquad \cdot \frac{n_{w_3}^{(h^-)} + I(w_2 = w_3) + \alpha_0 P_0(w_3)}{n^- + 1 + \alpha_0} \cdot \frac{n_u^{(h^-)} + I(u_{w_2} = u_{w_3}) + \frac{\rho}{2}}{n^- + 1 + \rho} \end{aligned} \tag{17}$$

where $I(.)$ is an indicator function taking on the value 1 when its argument is true, and 0 otherwise. The $I(.)$ terms, and the extra 1 in the denominators of the third and fourth factors, account for the fact that when generating w_3, the conditioning context consists of h^- plus one additional word and boundary location.

After initializing word boundaries at random (or non-randomly; see experiments below), the Gibbs sampler iterates over the entire data set multiple times. On each iteration, every potential boundary point is sampled once using Equations 16 and 17. After the burn-in period, these samples will be approximately distributed according to the posterior distribution $P(h|d)$.

Our Gibbs sampler has the advantage of being straightforward to implement, but it also has the disadvantage that modifications to the current hypothesis are small and local. Consequently, mobility through the hypothesis space may be low, because movement from one hypothesis to another very different one may require transitions through many low-probability intermediate hypotheses. Since the initial random segmentation is unlikely to be near the high-probability part of the solution space, it may take a very long time for the algorithm to reach that part of the space.

To alleviate this problem and reduce convergence time, we modified the Gibbs sampler to use simulated annealing (Aarts and Korst, 1989). Annealing the sampler causes it to make low-probability choices more frequently early in search, which allows it to more rapidly explore a larger area of the search space. Annealing is achieved by using a *temperature* parameter γ that starts high and is gradually reduced to 1. Annealing with a temperature of γ corresponds to raising the probabilities in the distribution under consideration (in this case, h_1 and h_2) to the power of $\frac{1}{\gamma}$ prior to sampling. Thus, when $\gamma > 1$, the sampled distribution becomes more uniform, with low-probability transitions becoming more probable. As the temperature is reduced, samples become more and more concentrated in the high-probability areas of the search space. Notice also that if the temperature is reduced below 1, the sampled distribution becomes

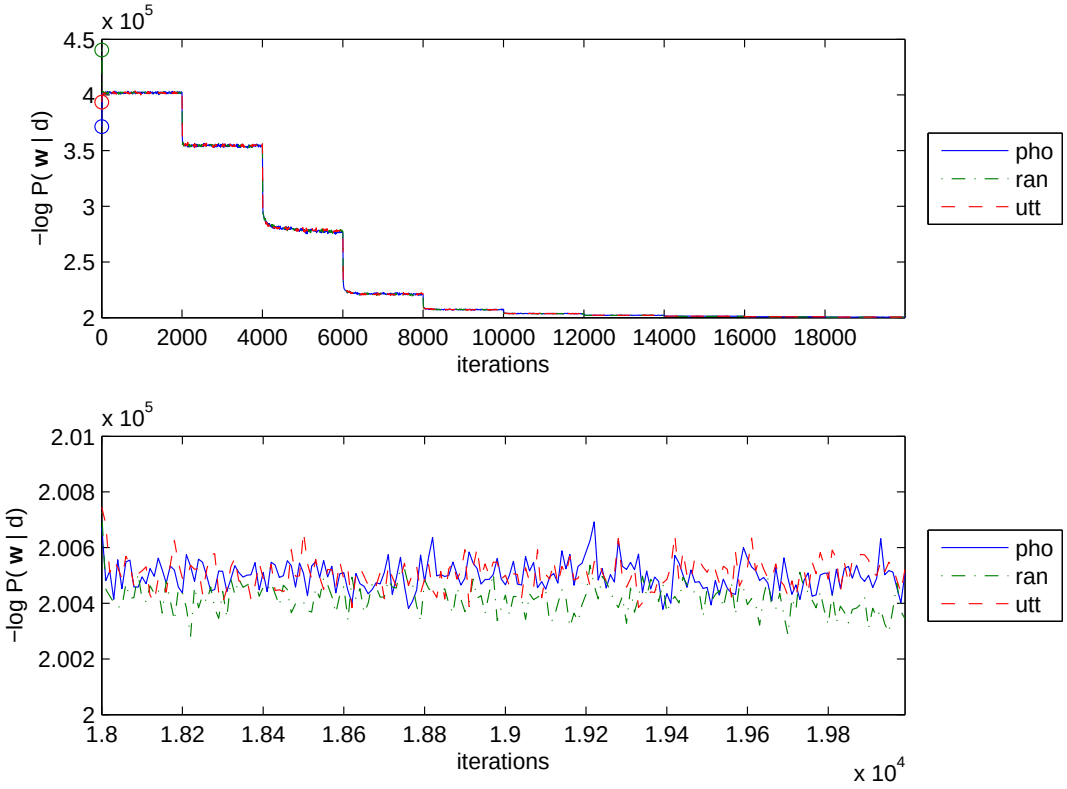

Figure 12: Trace plots of the posterior probabilities of samples from simulations for the DP model initialized with a boundary after every phoneme ('pho'), with random boundaries ('ran'), and with a boundary only at the end of each utterance ('utt'). Top: trace plot for the entire run of the algorithm, plotted every 10 iterations. The initial probabilities of each run (circles at $x = 0$) are very different, but within a few iterations the plots are barely distinguishable. Steep drops in the plots occur when the temperature is lowered. Bottom: detail of the final part of the plot, showing the overlap between the three simulations.

even more peaked, so that in the limit as $\gamma \to 0$, all probability mass will be concentrated on the mode of the distribution. This means that, by reducing the temperature to almost zero, we can obtain an approximation to the MAP solution.

All of the results in this paper are based on one of three possible annealing regimes. For most of our simulations, we ran the sampler for 20,000 iterations, annealing in 10 increments of 2000 iterations each, with $\frac{1}{\gamma} = (.1, .2, \ldots, .9, 1)$. Trace plots of three simulations using different initializations can be found in Figure 12, illustrating the effects of the annealing process on the posterior and the fact that different initializations make no difference to the final results. For simulations in which we wished to evaluate using the MAP approximation, we extended the run of the sampler for an additional 40,000 iterations, multiplying the temperature by 5/6 after every 2000 of these iterations.

Our third annealing regime was used to compare the probabilities assigned by the DP model and MBDP to a wide range of segmentations. These segmentations were generated by running the sampler for 50,000 iterations, slowly decrementing the temperature in 500 increments, with

$\frac{1}{\gamma} = (.002, .004, \ldots, .998, 1)$). A single sample was taken at each temperature.

A.2 Bigram model

A.2.1 The hierarchical Dirichlet process

To extend our model to include bigram dependencies, we use a *hierarchical Dirichlet process* (HDP) (Teh et al., 2005). This approach is similar to previously proposed n-gram models using hierarchical Pitman-Yor processes (Goldwater et al., 2006b; Teh, 2006). The HDP is a model that can be used in situations in which there are multiple distributions over similar sets of outcomes, and the distributions are believed to be similar. For language modeling, we can define a bigram model by assuming each word has a different distribution over the words that follow it, but all these distributions are linked. The definition of the HDP bigram language model (disregarding utterance boundaries for the moment) is

$$\begin{aligned} w_i \,|\, w_{i-1} = \ell,\ H_\ell &\sim H_\ell && \forall \ell \\ H_\ell \,|\, \alpha_1, G &\sim \mathrm{DP}(\alpha_1, G) && \forall \ell \\ G \,|\, \alpha_0, P_0 &\sim \mathrm{DP}(\alpha_0, P_0) \end{aligned}$$

That is, $P(w_i \,|\, w_{i-1} = \ell)$ is distributed according to H_ℓ, a DP specific to lexical item ℓ. H_ℓ is linked to the DPs for all other words by the fact that they share a common base distribution G, which is generated from another DP.

As in the unigram model, H_ℓ and G are never represented explicitly. By integrating over them, we get a distribution over bigram frequencies that can be understood in terms of the CRP, as illustrated in Figure 13. Each lexical item ℓ is associated with its own restaurant, which represents the distribution over words that follow ℓ. Different restaurants are not completely independent, however: the labels on the tables in the restaurants are all chosen from a common base distribution, which is represented by another CRP. A word ℓ' that has high probability in the base distribution will tend to appear in many different bigram types (i.e. following many other word types). However, $P(\ell' \,|\, \ell)$ may be very different for different ℓ, since each ℓ has its own restaurant for bigram counts.

To understand how our bigram model accounts for utterance boundaries, it is easiest if we consider the utterance boundary marker \$ as a special word type, so that w_i ranges over $\Sigma^+ \cup \{\$\}$. To generate an utterance, each word is chosen in sequence, conditioned on the previous word, until an utterance boundary is generated. The predictive probability distribution over the ith word is

$$\begin{aligned} P_2(w_i \,|\, \mathbf{w}_{-i}, \mathbf{z}_{-i}) &= \int P(w_i \,|\, H_{w_{i-1}}) P(H_{w_{i-1}} \,|\, \mathbf{w}_{-i}, \mathbf{z}_{-i})\ dH_{w_{i-1}} \\ &= \frac{n_{\langle w_{i-1}, w_i \rangle} + \alpha_1 P_1(w_i \,|\, \mathbf{w}_{-i}, \mathbf{z}_{-i})}{n_{w_{i-1}} + \alpha_1} \end{aligned} \tag{18}$$

where $n_{\langle w_{i-1}, w_i \rangle}$ is the number of occurrences of the bigram $\langle w_{i-1}, w_i \rangle$ in $\mathbf{w}_{-i}$ (we suppress the superscript $\mathbf{w}_{-i}$ notation) and $P_1(w_i \,|\, \mathbf{w}_{-i}, \mathbf{z}_{-i})$ is defined as

$$\begin{aligned} P_1(w_i \,|\, \mathbf{w}_{-i}, \mathbf{z}_{-i}) &= \int P(w_i \,|\, G) P(G \,|\, \mathbf{w}_{-i}, \mathbf{z}_{-i})\ dG \\ &= \frac{t_{w_i} + \alpha_0 P_0'(w_i)}{t + \alpha_0} \end{aligned} \tag{19}$$

where t_{w_i} is the total number of bigram tables (across all words) labeled with w_i, t is the total number of bigram tables, and P_0' is defined to allow generation of either an utterance boundary

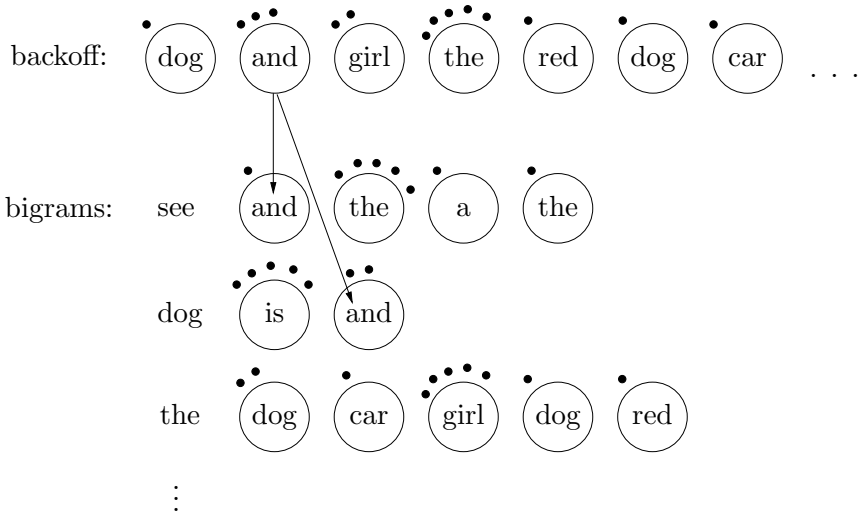

Figure 13: Bigrams are modeled using a hierarchical Chinese restaurant process. Each lexical item ℓ has its own restaurant to represent the distribution of tokens following ℓ in the data. The labels on the tables in these bigram restaurants are drawn from the distribution in the backoff or "master" restaurant (top). Each customer (black dot) in the bigram restaurants represents a bigram token; each customer in the backoff restaurant represents a label on some bigram table.

or a string of phonemes $x_1 \dots x_M$:

$$P_0'(w_i) = \begin{cases} p_\$ & w_i = \$ \\ (1 - p_\$) P_0(w_i) & w_i \in \Sigma^+ \end{cases} \tag{20}$$

where $p_\$$ is a parameter of the model, and P_0 is the lexical model used in the unigram model.[14]

In the model just defined, P_1 is the posterior estimate of the base distribution shared by all bigrams, and can be viewed as a unigram backoff. In P_1, words are generated from the DP G. Since G determines the probability that a word type appears on a bigram table, P_1 is estimated from the number of tables on which each type appears. In other words, when a particular bigram sequence $\langle w_{i-1}, w_i \rangle$ is never observed in $\mathbf{w}_{-i}$, the probability of w_i following w_{i-1} is estimated using the number of different word types that have been observed to precede w_i. If this number is high, then $P(w_i|w_{i-1})$ will be higher than if this number is low.[15]

A.2.2 Inference

Inference can be performed on the HDP bigram model using a Gibbs sampler similar to the sampler used for the unigram model. Our unigram sampler relied on the fact that words in

[14]Technically, allowing P_0 to generate utterance boundaries regardless of context permits our model to generate two consecutive utterance boundaries (i.e., an utterance with no words). An earlier version of the model (Goldwater et al., 2006a; Goldwater, 2006) disallowed empty utterances, but was slightly more complicated. Since we use the model for inference only, we have adopted the simpler assumption here.

[15]Many standard n-gram smoothing methods use similar kinds of estimates based on both type and token counts. In fact, Kneser-Ney smoothing (Kneser and Ney, 1995), a particularly effective smoothing technique for n-gram models (Chen and Goodman, 1998), has been shown to fall out naturally as the posterior estimate in a hierarchical Bayesian language model similar to the one described here, with the DPs replaced by Pitman-Yor processes (Goldwater et al., 2006b; Teh, 2006).

the unigram model were exchangeable, so that (for example) $P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) = P(w_3)P(w_1 \mid w_3)P(w_2 \mid w_3, w_1)$. While our bigram model is clearly not exchangeable at the level of words, it *is* exchangeable at the level of bigrams. To generate a sequence of words such as $w_1w_2w_3$, four bigrams must be generated: $\langle \$, w_1 \rangle$, $\langle w_1, w_2 \rangle$, $\langle w_2, w_3 \rangle$, and $\langle w_3, \$ \rangle$. Under the HDP model, the order in which those bigrams is generated does not affect their joint probability. This property allows us to sample one boundary at a time in much the same way we did in the unigram model. Since we must now consider context during sampling, we write the two possible segmentations as $s_1 = \beta w_l w_1 w_r \gamma$ and $s_2 = \beta w_l w_2 w_3 w_r \gamma$, with w_l and w_r being the left and right context words for the area under consideration, and β and γ being the remaining words. A complicating factor that did not arise in the unigram model is the fact that bigram probabilities depend on the seating assignment of the words under consideration. (The seating assignment affects the the number of tables assigned to each word, which is used to compute the unigram probability P_1.) Each hypothesis h therefore consists of a segmentation of the data, along with an assignment to tables of the words in that segmentation. We use $\mathcal{H}_1$ to refer to the set of hypotheses consistent with s_1, $\mathcal{H}_2$ to refer to the set of hypotheses consistent with s_2, and h^- to refer to the set of bigrams and table assignments shared by $\mathcal{H}_1$ and $\mathcal{H}_2$ (i.e., the bigrams covering βw_l and $w_r \gamma$, plus the table assignments for those words). Then we have

$$
\begin{aligned}
&P(h \mid h^-) \\
&\quad = \begin{cases}
P(\langle w_l, w_1 \rangle \mid h^-) \cdot P(\langle w_1, w_r \rangle \mid \langle w_l, w_1 \rangle, h^-) \cdot P(z_1 \mid \langle w_l, w_1 \rangle, h^-) & h \in \mathcal{H}_1 \\
P(\langle w_l, w_2 \rangle \mid h^-) \cdot P(\langle w_2, w_3 \rangle \mid \langle w_l, w_2 \rangle, h^-) \cdot P(\langle w_3, w_r \rangle \mid \langle w_l, w_2 \rangle, \langle w_2, w_3 \rangle, h^-) & \\
\quad \cdot P(z_2 \mid \langle w_l, w_2 \rangle, h^-) \cdot P(z_3 \mid z_2, \langle w_2, w_3 \rangle, h^-) & h \in \mathcal{H}_2
\end{cases}
\end{aligned}
$$

where z_1, z_2, and z_3 are random variables representing the table numbers of w_1, w_2, and w_3.

Using these equations, we could compute the probability of every hypothesis in the set $\mathcal{H}_1 \cup \mathcal{H}_2$, and sample from this set. Instead, we implemented a simpler and more efficient sampler by adopting the following approximations (which are exact provided there are no repeated unigrams or bigrams in either $\{w_l, w_1, w_r\}$ or $\{w_l, w_2, w_3, w_r\}$):[16]

$$
P(h \mid h^-) \;=\; \begin{cases}
P(\langle w_l, w_1 \rangle \mid h^-) \cdot P(\langle w_1, w_r \rangle \mid h^-) \cdot P(z_1 \mid \langle w_l, w_1 \rangle, h^-) & h \in \mathcal{H}_1 \\
P(\langle w_l, w_2 \rangle \mid h^-) \cdot P(\langle w_2, w_3 \rangle \mid h^-) \cdot P(\langle w_3, w_r \rangle \mid h^-) & \\
\quad \cdot P(z_2 \mid \langle w_l, w_2 \rangle, h^-) \cdot P(z_3 \mid \langle w_2, w_3 \rangle, h^-) & h \in \mathcal{H}_2
\end{cases}
$$

These approximations allow us to sample a hypothesis from $\mathcal{H}_1 \cup \mathcal{H}_2$ in two steps. First, we decide whether our hypothesis will be from $\mathcal{H}_1$ or $\mathcal{H}_2$ (i.e., whether the segmentation will be s_1 or s_2). Then, we sample table assignments for either w_1 or w_2 and w_3, as appropriate. By assumption of the approximation, the assignments for w_2 and w_3 can be sampled independently.

More precisely, we sample a segmentation s using the following equations:

$$
P(s = s_1 \mid h^-) \;=\; \frac{n^{(h^-)}_{\langle w_l, w_1 \rangle} + \alpha_1 P_1(w_1 \mid h^-)}{n^{(h^-)}_{w_l} + \alpha_1} \cdot \frac{n^{(h^-)}_{\langle w_1, w_r \rangle} + \alpha_1 P_1(w_r \mid h^-)}{n^{(h^-)}_{w_1} + \alpha_1} \tag{21}
$$

$$
P(s = s_2 \mid h^-) \;=\; \frac{n^{(h^-)}_{(w_l, w_2)} + \alpha_1 P_1(w_2 \mid h^-)}{n^{(h^-)}_{w_l} + \alpha_1} \cdot \frac{n^{(h^-)}_{(w_2, w_3)} + \alpha_1 P_1(w_3 \mid h^-)}{n^{(h^-)}_{w_2} + \alpha_1} \cdot \frac{n^{(h^-)}_{(w_3, w_r)} + \alpha_1 P_1(w_r \mid h^-)}{n^{(h^-)}_{w_3} + \alpha_1} \tag{22}
$$

[16] It would be straightforward to add a Metropolis-Hastings correction step (Metropolis et al., 1953; Hastings, 1970; Neal, 1993) to correct for the slight discrepancy between our approximation and the true distribution $P(h \mid h^-)$. However, we expect the number of cases in which our approximation is incorrect to be small, and the difference between the approximate and true distributions to be slight, so we did not implement the correction step.

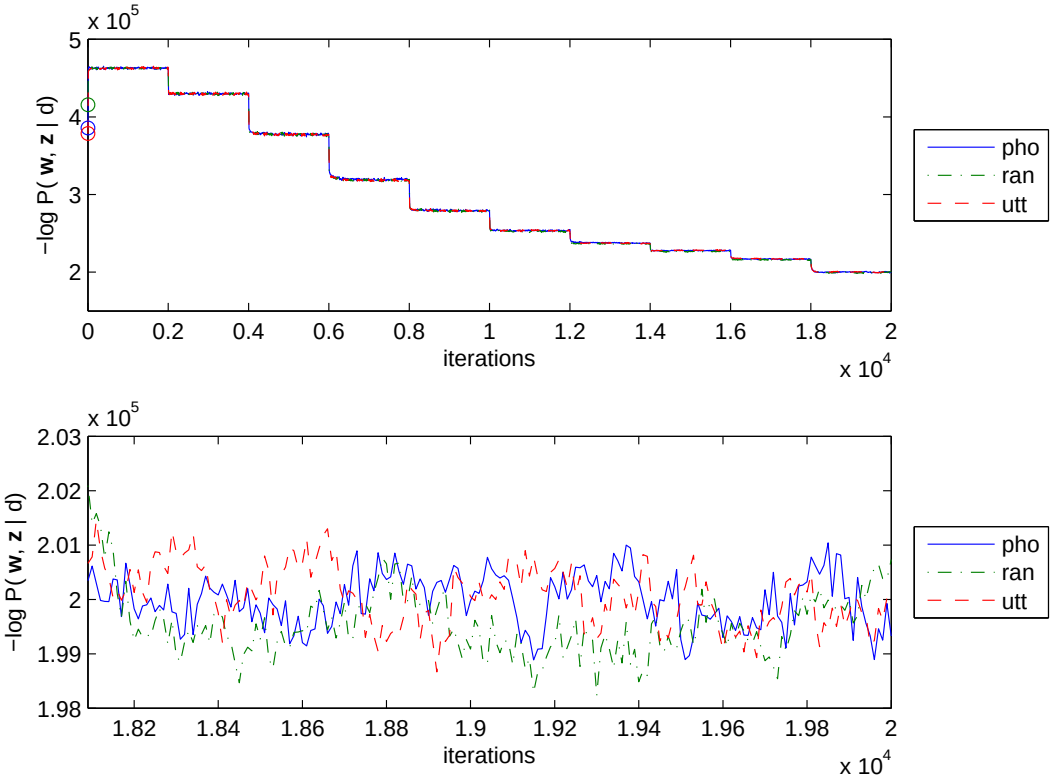

Figure 14: Trace plots of the posterior probabilities of samples from samplers for the HDP model initialized with a boundary after every phoneme ('pho'), with random boundaries ('ran'), and with a boundary only at the end of each utterance ('utt'). Top: trace plot for the entire run of the algorithm, plotted every 10 iterations, with the initial probabilities of each run circled at $x = 0$. Bottom: detail of the final part of the plot.

Given w_i and h^-, z_i has the following distribution:

$$P(z_i = k \,|\, w_i, h^-) \propto \begin{cases} n_k^{(h^-)} & 1 \le k \le K(w_i^{(h^-)}) \\ \alpha P_1(w_i) & k = K(w_i^{(h^-)}) + 1 \end{cases} \tag{23}$$

where $K(w_i^{(h^-)})$ is the number of tables assigned to w_i in h^-. After sampling s, we use Equation 23 to sample values for either z_1 or each of z_2 and z_3.

We used the same annealing regimes described above to encourage faster convergence of our bigram sampler and obtain an approximation to the MAP segmentation. Figure 14 illustrates that, as in the unigram sampler, three different simulations with different initializations end up producing samples with similar posterior probabilities. We plot the joint probability $P(\mathbf{w}, \mathbf{z})$ since each sample is an assignment to both words and tables; the column labeled $-\log P(\mathbf{w})$ in Table 8 in the main text actually gives $-\log P(\mathbf{w}, \mathbf{z})$ as well, but we omitted the $\mathbf{z}$ from the label since there is no mention of table assignments in our exposition of the model there.

Note that our sampler differs somewhat from the original sampler described for this model (Goldwater et al., 2006a), which used an approximation that did not explicitly track assignments

to bigram tables. The difference in performance between the approximate version and the version presented here is minimal; results reported here are different from previously reported results primarily due to fixing a small bug that was discovered while making the other modifications to our sampler.

B Proofs

This Appendix presents two proofs. First, we show that there is a generative procedure for the DP model that is in the same form as that of MBDP-1, allowing us to show that these procedures make identical assumptions about two of the distributions used in generating segmented corpora, and then we show that one of these distributions – the distribution on orderings – has to be used in the generative procedure assumed by any unigram model.

B.1 Connecting MBDP-1 and the DP model

The probabilistic model behind MBDP-1 is based on a sequence of four steps that are intended to generate a segmented corpus. The steps are as follows:

Step 1 Sample the number of word types, K, from a distribution $P(K)$.

Step 2 Sample a vector of frequencies $\mathbf{n} = (n_1, \ldots, n_K)$ for these words from a distribution $P(n_k|K)$.

Step 3 Sample the lexicon $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_K)$ from a distribution $P(\boldsymbol{\ell}|K)$.

Step 4 With $n = \sum_j n_j$ being the total number of word tokens, sample an ordering $s : \{1, \ldots, n\} \rightarrow \{1, \ldots, K\}$, constrained to map exactly n_j positions to word ℓ_j, from a distribution $P(s|\mathbf{n})$. The words in the corpus are $\mathbf{w} = (w_1, \ldots, w_n) = (\ell_{s(1)}, \ldots, \ell_{s(m)})$.

If we follow this procedure, the probability of a particular segmented corpus, $\mathbf{w}$, is

$$P(\mathbf{w}) = \sum_{K, \boldsymbol{\ell}, \mathbf{n}, s} P(\mathbf{w}|s, \mathbf{n}, \boldsymbol{\ell}, K) P(s|\mathbf{n}) P(\mathbf{n}|K) P(\boldsymbol{\ell}|K) P(K) \tag{24}$$

where $P(\mathbf{w}|s, \mathbf{n}, \boldsymbol{\ell}, K) = 1$ if the other variables generate $\mathbf{w}$ and 0 otherwise. If all frequencies in $\mathbf{n}$ are non-zero, then for any given value of $\mathbf{w}$, only one set of (lexical item, frequency) pairs is consistent with $\mathbf{w}$, so the sum in Equation 24 simply counts the $K!$ permutations of the indices of those (ℓ_j, n_j) pairs, each of which has the same probability under the generative process.

MBDP-1 uses a particular distribution in each of the four steps. In Step 1, the number of word types is drawn from $P(K) \propto 1/K^2$. In Step 2, frequencies are assumed to be independent, with $P(\mathbf{n}|K) = \prod_{j=1}^{K} P(n_j)$ where $P(n_j) \propto 1/n_j^2$. In Step 3, words are generated by choosing phonemes independently, with $P(\ell) = \frac{1}{1-p_\#} \prod_{k=1}^{q} P(x_k)$, where q is the number of phonemes in ℓ, x_k is the kth phoneme (which might be the end marker, #) and $P(x)$ is a probability distribution over phonemes. The words in the lexicon are required to be unique, otherwise the possible configurations of K, $\boldsymbol{\ell}$, $\mathbf{n}$ and s that could have produced $\mathbf{w}$ proliferate.[17] In Step 4, $P(s|\mathbf{n})$ is taken to be uniform over all $\binom{n}{n_1 \ldots n_k}$ valid mappings.

The probabilistic model used in MBDP has the unusual property of generating a corpus as a single object, without specifying how a sequence of words might be generated one after another. This property makes it difficult to form predictions about what the next word or utterance might be, and to compute conditional distributions. Indeed, in the description of MBDP-1 it is noted,

[17] We maintain the unique lexicon requirement for the models discussed in this appendix since it avoids having to sum over many solutions in computing $P(\mathbf{w})$. However, our equivalence results hold provided the distribution from which the lexicon is drawn, $P(\boldsymbol{\ell}|K)$, is exchangeable, with the probability of $\boldsymbol{\ell}$ not depending on the indices of the ℓ_j. A relatively complex (but exchangeable) distribution based on $P(\ell)$ was used in Brent (1999).

when referring to the ratio of the probability of one corpus to another that "...the semantics of the model are not consistent with a conditional probability interpretation. The sequence $w_1, \ldots, w_k$ is not a conjunction of events from the probability space but rather a single event that is determined by the joint outcomes of steps 1-4 above. Thus, $w_1, \ldots, w_{k-1}$ and $w_1, \ldots, w_k$ are actually distinct, mutually exclusive events from the probability space." (Brent (1999), p. 104). In contrast, the DP model is defined in terms of a series of conditional distributions, with each word being sampled conditioned on the previous words. In the remainder of this section, we will show that the DP model can nonetheless be specified by a procedure similar to that used in MBDP-1.

We will start by assuming that in Step 1 we fix the number of word types, K. In Step 2, we draw each n_j from a Poisson(λ) distribution where λ is generated from a Gamma($\frac{\alpha}{K}, \beta$) distribution. Integrating over λ, this gives

$$\begin{aligned} P(n_j) &= \int_0^\infty P(n_j|\lambda)p(\lambda)\ d\lambda & (25) \\ &= \frac{1}{n_j!}\frac{\beta^{\frac{\alpha}{K}}}{\Gamma(\frac{\alpha}{K})}\int_0^\infty \exp\{-\lambda(1+\beta)\}\lambda^{n_j+\frac{\alpha}{K}-1}\ \mathrm{d}\lambda \\ &= \frac{1}{n_j!}\frac{\beta^{\frac{\alpha}{K}}}{\Gamma(\frac{\alpha}{K})}\frac{\Gamma(n_j+\frac{\alpha}{K})}{(1+\beta)^{n_j+\frac{\alpha}{K}}} & (26) \end{aligned}$$

where the neat result is due to conjugacy between the Gamma and Poisson. As a result, we have

$$\begin{aligned} P(\mathbf{n}|K) &= \prod_{j=1}^{K}\frac{1}{n_j!}\frac{\beta^{\frac{\alpha}{K}}}{\Gamma(\frac{\alpha}{K})}\frac{\Gamma(n_j+\frac{\alpha}{K})}{(1+\beta)^{n_j+\frac{\alpha}{K}}} \\ &= \left(\frac{\alpha}{K}\right)^{K_+}\left(\frac{\beta}{1+\beta}\right)^{\alpha}\prod_{\{j|n_j>0\}}\frac{1}{n_j!}\frac{1}{(1+\beta)^{n_j}}\frac{\Gamma(n_j+\frac{\alpha}{K})}{\Gamma(1+\frac{\alpha}{K})} & (27) \end{aligned}$$

where the notation on the product indicates that only words with non-zero frequencies should be included, and K_+ is the size of this set. We can expand out the factorial and Gamma functions, to give

$$P(\mathbf{n}|K) = \left(\frac{\alpha}{K}\right)^{K_+}\left(\frac{\beta}{1+\beta}\right)^{\alpha}\prod_{j=1}^{K_+}\left(\frac{1}{n_j}\frac{1}{(1+\beta)^{n_j}}\prod_{j=1}^{n_j-1}\frac{j+\frac{\alpha}{K}}{j}\right) \quad (28)$$

where we simplify the constraint on j by assuming that the first K_+ indices over which j quantifies correspond to words that have frequencies greater than zero. Steps 3 and 4 can proceed as in MBDP.

Applying Equation 24, the probability of a segmented corpus $\mathbf{w}$ results from summing over all configurations of $\boldsymbol{\ell}$, $\mathbf{n}$, and s that produce $\mathbf{w}$. We can marginalize out the words ℓ_j for which $n_j = 0$, so $P(\boldsymbol{\ell}) = P(\boldsymbol{\ell}_+)$, where $\boldsymbol{\ell}_+$ is the set of words with non-zero frequencies. There are then $\frac{K!}{K_0!}$ configurations of $\boldsymbol{\ell}$ and $\mathbf{n}$ that yield the same same corpus $\mathbf{w}$, where $K_0 = K - K_+$ is the number of words for which $n_j = 0$. This corresponds to the $\binom{K}{K_0}$ ways of allocating K_0 zeros across the K words, multiplied by the $K_+!$ permutations of the indices of (ℓ_j, n_j) pairs with $n_j > 0$. Finally, $P(s|n) = 1/\binom{n}{n_1 \ldots n_{K_+}}$. Putting all of this together with Equation 28, we obtain

$$P(\mathbf{w}) = P(\boldsymbol{\ell}_+)\frac{K!}{K_0!}\frac{\prod_{j=1}^{K_+}n_j!}{n!}\left(\frac{\alpha}{K}\right)^{K_+}\left(\frac{\beta}{1+\beta}\right)^{\alpha}\prod_{j=1}^{K_+}\left(\frac{1}{n_j}\frac{1}{(1+\beta)^{n_j}}\prod_{j=1}^{n_j-1}\frac{j+\frac{\alpha}{K}}{j}\right). \quad (29)$$

We are now in a position to generalize this distribution to the case where $K \to \infty$, with

$$\begin{aligned}\lim_{K\to\infty} P(\mathbf{w}) &= P(\boldsymbol{\ell}_+)\frac{\prod_{j=1}^{K_+} n_j!}{n!}\alpha^{K_+}\left(\frac{\beta}{1+\beta}\right)^{\alpha}\prod_{j=1}^{K_+}\frac{1}{n_j}\frac{1}{(1+\beta)^{n_j}} \\ &= P(\boldsymbol{\ell}_+)\frac{\alpha^{K_+}}{n!}\frac{\beta^{\alpha}}{(1+\beta)^{n+\alpha}}\prod_{j=1}^{K_+}(n_j-1)!, \qquad (30)\end{aligned}$$

where the result follows from the fact that $\frac{K!}{K_0!\ K^{K_+}} = \prod_{j=1}^{K^+}\frac{1}{K}(K-j-1)$ and $\lim_{K\to\infty}(\prod_{j=1}^{K^+}\frac{1}{K}(K-j-1))(\prod_{j=1}^{K_+}\prod_{k=1}^{n_j-1}\frac{k+\frac{\alpha}{K}}{k}) = 1$.

Equation 30 gives a distribution over segmented corpora derived via the MBDP-1 scheme of generating a corpus as a single object. However, we can also ask what the distribution over word sequences would be if we conditioned on the total number of words. To do this, we need to divide $P(\mathbf{w})$ by the probability of a particular value of n, $P(n)$, under this model. Fortunately, $P(n)$ is easy to compute. We chose $P(n_j)$ to be a Poisson distribution with rate set by a gamma distribution, and $n = \sum_{j=1}^{\infty} n_j$. The sum of a set of Poisson random variables is Poisson with rate equal to the sum of the rates of the original variables. The gamma distribution has a similar property, meaning that the sum of the rates of our Poisson distributions follows a Gamma(α, β) distribution. Summing out this variable as we did for the individual frequencies, we obtain

$$P(n) = \frac{1}{n!}\frac{\beta^{\alpha}}{(1+\beta)^{n+\alpha}}\frac{\Gamma(n+\alpha)}{\Gamma(\alpha)}. \qquad (31)$$

Dividing Equation 30 by $P(n)$ gives the conditional distribution over corpora given their length

$$P(\mathbf{w}|n) = P(\boldsymbol{\ell}_+)\alpha^{K_+}\frac{\Gamma(\alpha)}{\Gamma(n+\alpha)}\prod_{j=1}^{K_+}(n_j-1)! \qquad (32)$$

which is exactly the distribution over word sequences yielded by the DP model.

B.2 Relationship to other unigram models

The result in the previous section implies that the DP model can be specified in a way such that it uses exactly the same distributions as MBDP-1 in Steps 3 and 4. We can now generalize this result, showing that *any* unigram model has to use the same distribution in Step 4, taking a uniform distribution over orderings of word tokens. We do this by showing that exchangeability – assigning the same probability to all sequences of words in which the words occur with the same frequencies – implies the existence of a generative procedure in which we first choose frequencies according to some distribution, $P(\mathbf{n})$, and then choose an ordering of words uniformly at random. Exchangeability is the essence of the assumption behind a unigram model, since it indicates that we are completely indifferent to the ordering of the words, and must be a property of all unigram models.

For any distribution over segmented corpora $\mathbf{w}$, we can compute $P(\mathbf{n})$ by summing over all $\mathbf{w}$ in which the frequencies are $\mathbf{n}$. Exchangeability means that all permutations of the indices of words in the corpus have the same value, and permutation of indices does not affect $\mathbf{n}$. There is also a unique $\mathbf{n}$ for any sequence. Consequently, $P(\mathbf{n})$ will be the probability of any single sequence with frequencies corresponding to $\mathbf{n}$ (which will be the same for all such sequences), multiplied by the number of such sequences. Let $m_{\mathbf{n}}$ represent the number of unique sequences yielded by frequencies $\mathbf{n}$ and $\mathbf{w}^*_{\mathbf{n}}$ be an arbitrary sequence with these frequencies. We then have $P(\mathbf{n}) = m_{\mathbf{n}}P(\mathbf{w}^*_{\mathbf{n}})$. Now, we can compute $P(\mathbf{w}|\mathbf{n})$, the distribution over segmented corpora given frequencies. This will just be $P(\mathbf{w})$ divided by $P(\mathbf{n})$. Since $P(\mathbf{w})$ is the same as $P(\mathbf{w}^*_{\mathbf{n}})$ by

exchangeability, this reduces to $1/m_{\mathbf{n}}$ for any corpus $\mathbf{w}$. This is equivalent to simply choosing uniformly at random from all valid permutations, as $m_{\mathbf{n}}$ is just $\binom{n}{n_1 \ldots n_k}$.